Ball 4 on Forensics

Four Articles on Computer Forensics for Lawyers

by Craig Ball

Computer Forensics for Lawyers Who Can't Set a Digital Clock

Meeting the Challenge: E-Mail in Civil Discovery

Finding the Right Computer Forensics Expert

Cross Examination of the Computer Forensics Expert

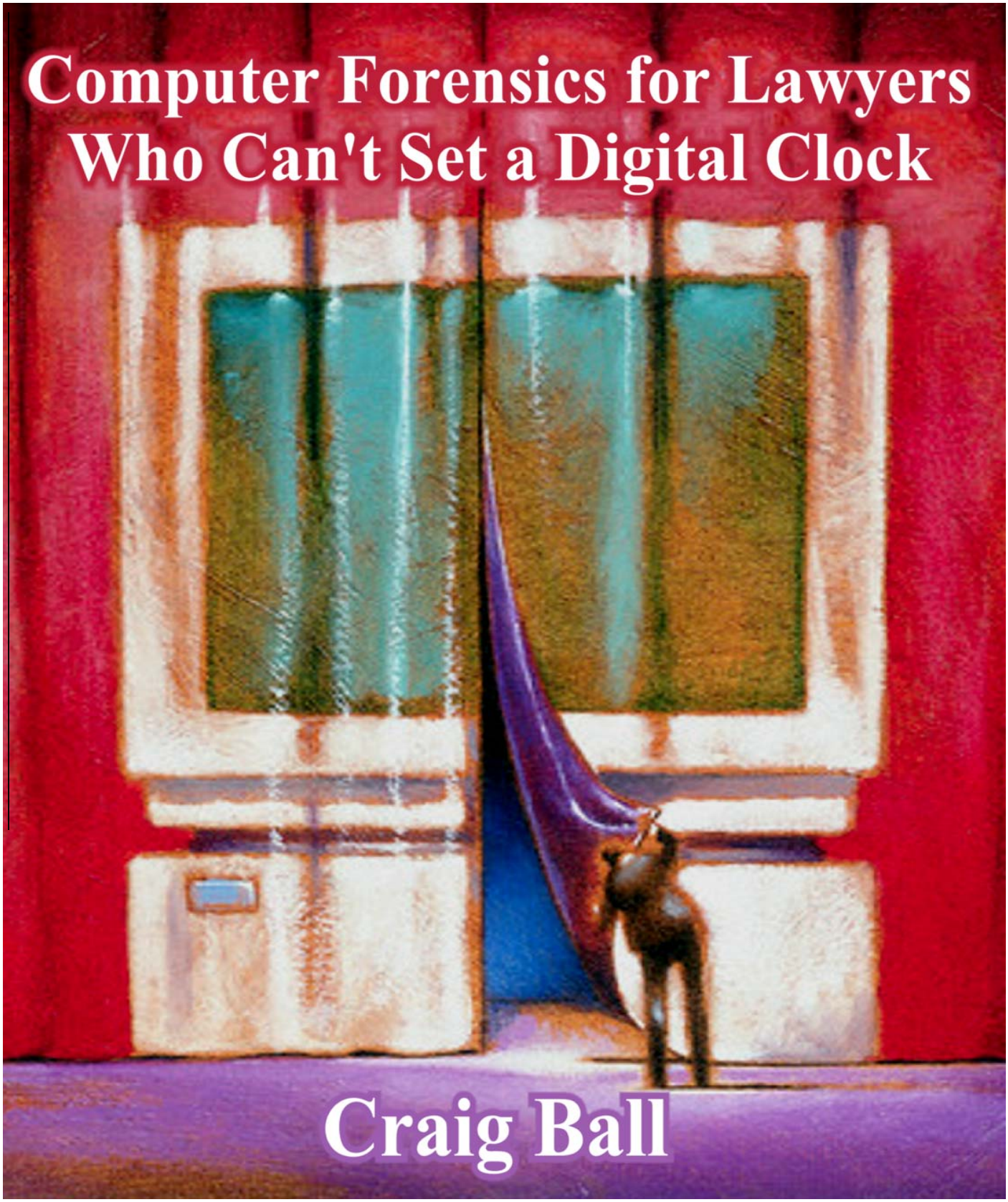
Four on Forensics

Four Articles on Computer Forensics for Lawyers

Everyone uses computers—at home, at work, on the road, leaving voicemail, opening card key doors--everywhere, every day. Nearly all documentary evidence is created digitally, and only about a third or less gets printed out. As lawyers, we're duty bound to zealously pursue the truth, so we can't walk away from 2/3rds of the evidence or turn a blind eye to its metadata. We must master electronic discovery and learn to exploit its powerful sub-discipline, computer forensics.

This quartet of articles introduce tech-challenged litigators to computer forensics and offer a host of practical strategies geared to helping you win your cases with the power of computer forensics and electronic discovery.

Contents:



Computer Forensics for Lawyers Who Can't Set a Digital Clock

Table of Contents

Note to Readers:

This article focuses on technical matters impacting the cost, complexity and scope of e-discovery, rather than the burgeoning case law. For extensive resources on electronic discovery law, please look at other materials available at www.craigball.com and visit the following helpful sites:

K&L Gates Electronic Discovery Law Site
http://www.ediscoverylaw.com/
Berkman Center for Internet & Society at Harvard Law School
http://cyber.law.harvard.edu/digitaldiscovery/library.html
Discovery Resources
http://discoveryresources.org/

For extensive links to further information about computer forensics, visit:
The Electronic Evidence Information Center
http://www.e-evidence.info/index.html

Computer Forensics for Lawyers Who Can't Set a Digital Clock

"When you go looking for something specific, your chances of finding it are very bad. Because of all the things in the world, you're only looking for one of them. When you go looking for anything at all, your chances of finding it are very good. Because of all the things in the world, you're sure to find some of them."

Movie Detective Daryl Zero, from the film "The Zero Effect"

The Smoking Gun

Lawyers love the smoking gun. We adore the study that shows it's cheaper to pay off the burn victim than fix the flawed fuel system, the directive that staff needs to work all night to implement the new document "retention" policy, the employment review with the racist remark and the letter between competitors agreeing to "respect" each other's pricing. Each case has its smoking gun. It may be a peashooter with the faintest whiff of cordite or a Howitzer with a red-hot muzzle, but it's there *somewhere*. Searching for the smoking gun once meant poring over great forests felled, turned to oceans of paper captured in folders, boxes, cabinets, rooms and warehouses. Today, fewer and fewer business communications and records find their way into paper form, so your smoking gun is likely smoking on someone's hard drive.

What's more, not only is the smoking gun more likely to be stored electronically, the informal and immediate nature of electronic communications makes them more likely to be smoking guns. People aren't as guarded in what they say via e-mail as when writing a letter. Electronic communication is so frictionless that a damning e-mail is just an improvident click away from dozens or hundreds or thousands of in boxes. Think also of the ease of digitally distributing attachments that would have consumed hours at a copier to send on paper.

Consider also the volume of electronic communications. On a given day, I might send out fifty to one hundred individual e-mails, but it's unlikely I've drafted and sent that many letters in any day of my entire career as an attorney. Put another way, I'm about fifty times more likely to put my foot in my mouth electronically than on paper. This is fast becoming the norm in American business.

What You Don't Know *Can* Hurt You

Although lawyers are coming to appreciate that the smoking gun they seek may not be on paper, a pervasive lack of knowledge about electronic data, coupled with experience grounded exclusively on paper discovery, makes it hard for lawyers and judges to meet the challenge of digital data discovery.

In a case involving a dispute over privileged documents on a shared laptop computer, the parties entered into an agreed order respecting the data on the computer, and the court appointed me as Special Master to carry out the tasks ordered. The instructions I received were simple…and daunting. Among other tasks, I was to reduce all "documents" on the computer to written form, including all scans, program files, deleted records and data from Internet surfing. Using round numbers, the hard drive in question had some ten gigabytes of data spread across 18,000 files. The way the assignment was structured, each file constituted a document and file

sizes ran the gamut from virtually nothing to massive programs. Because of the sensitive nature of the information, I was expected to personally handle all aspects of the task, including monitoring the printing.

Estimates of how digital data convert to printed pages are notoriously misleading because of the wide variance in how applications format the printed page: a tiny Word file can consume dozens of printed pages while a large graphic file may result in a small image. However, a commonly cited estimate suggests the following correlation:

Data	**Printed Pages**
One megabyte	= 1,000-1,400
One gigabyte	= 100,000-140,000
One terabyte	=100,000,000-140,000,000

By this measure, the ten gigabytes of data on the hard drive would print out to something over a million pages, and I could get the job done in under a year of forty-hour weeks, chained to the printer. Problem was, even if I were willing to abandon my practice and baby-sit a laser printer, the files were not formatted so as to efficiently fill the printed pages. Instead, I was probably looking at several million printed pages, the vast majority of them containing meaningless strings of gibberish. Did I mention I'd have to make three copy sets? The paper and toner alone would cost $120,000, not to mention the printers and Prozac.

Clearly, a global order that the contents of a computer be printed out is a disaster. The solution in this case was to revise the order to permit production of the data on CD-ROM in its native electronic format and to eliminate the production of software applications and other data that did not, in any manner, reflect activities by users of the computer. This is a much more time- and cost-efficient technique, and it spared a couple of acres of forest to boot.

A Little Knowledge is a Wonderful Thing

Errors like the potentially costly one just described can be avoided in the first place if lawyers gain a fundamental understanding of how a computer stores data and the many nooks and crannies where data can hide despite efforts to make it disappear. This knowledge is valuable whether you are combing an employee's computer to find out if they have engaged in on-the-job shenanigans with firm property or framing discovery requests; but be advised that it is no substitute for the services of a qualified and experienced computer forensics expert. If you don't know what you are doing, your efforts to resurrect deleted data may end up permanently deleting the smoking gun or, at the very least, imperiling its admissibility in court.

Reading this article isn't going to make you a computer forensics expert. Many topics are oversimplified or explained with metaphors that would make a computer engineer wince, but you will get enough of the basics to impress opposing counsel and make yourself wholly unattractive to members of the opposite sex. You might even find yourself casting admiring glances at short sleeve shirts and vinyl pocket protectors.

> *A little knowledge that acts is worth infinitely more than much knowledge that is idle.*
> ***-Kahlil Gibran, "The Prophet"***

This article will focus on the WinTel platform (geek speak for an Intel Pentium processor computer running the Microsoft Windows operating system), but all of the concepts and many of the specifics apply to other computing environments as well.

Magnetic Storage

A variety of technologies have to come together to create a computer, but the most important of these with respect to forensics has to be magnetic storage. Nearly all of the smoking gun data you seek to discover or shield from disclosure takes the forms of trillions upon trillions of faint and impossibly tiny magnetic charges that coat the surface of a rapidly spinning disc. A Lilliputian device, called a *read/write head,* interacts with these particles, imparting a magnetic charge or reading a charge already there. No matter what form information takes when it goes into a computer—video, sound, word, number, or photograph—it is all stored magnetically in a sequence of magnetic polarity changes customarily represented by ones and zeros. These "on" and "off" states are like the Morse code used by telegraphers one hundred fifty years ago, but now transmitted so quickly that an encyclopedia of information can be communicated in seconds.

It's Time

Can a lawyer be a damn good litigator without knowing much about the inner workings of a computer? Ten years ago, the answer would have been, "sure;" but we've reached the point where not understanding computer forensics and not having digital discovery skills is no laughing matter. It's a ticking time bomb in your practice. You *know* how important discovery is to winning your case. You know the value of the smoking gun document, the doctored record, and the too-candid memo. Products liability cases, wrongful discharge claims and antitrust actions, just to name a few, are won and lost in discovery. Try this fact on for size:

***Ninety-five percent* of the world's information is being generated and stored in digital form and few *business documents created today ever become paper records*. They *never* get printed out. They *never* leave the digital domain. Most *never* find their way into the printed material produced to you in discovery.**

Now ponder these questions:

Are you willing to accept an assurance of "we didn't find anything" from the other side when you know they haven't looked everywhere and they don't know how to find what they are supposed to be looking for?

Can you effectively cross-examine a computer expert if you know almost nothing about their area of expertise? How will you know when they are wrong? How can you expose their weaknesses?

Are you content to have to hire an expert in every case where computer records are at issue? And isn't that almost every case nowadays?

If the answer to any of these questions is "no," it's time to stop leaving the geek stuff to the geeks. It's time to learn the basics of computer forensics.

How Much Information?
The world produces between 1 and 2 exabytes of unique information per year, which is roughly 250 megabytes for every man, woman, and child on earth. An exabyte is a billion gigabytes, or 10^{18} bytes, *equivalent to the textual content of a trillion books*. Printed documents of all kinds comprise only .003% of the total. Magnetic storage is by far the largest medium for storing information and is the most rapidly growing, with shipped hard drive capacity *doubling every year.*

Single hard drives now hold a gigabyte of data and sell for less than forty cents per gigabyte, a two-thousand-fold price drop in just a few years time. By way of comparison, if the automobile industry were as efficient, you could buy a new car for less than you paid for your last haircut!

Computer Forensics
Computer forensics is the identification, preservation, extraction, interpretation and presentation of computer-related evidence. It sounds like something anyone who knows his way around a computer might be able to do, and in fact, many who offer their services as computer forensic specialists have no formal forensic training or certification--which is not to say they can't do the job well, but it certainly makes it hard to be confident they can! There are compelling reasons to hire a formally trained and experienced computer forensic specialist. Far more information is retained by a computer than most people realize, and without using the right tools and techniques to preserve, examine and extract data, you run the risk of losing something important, rendering what you do find inadmissible, or even being charged with spoliation of the evidence.

The cardinal rules of computer forensics can be expressed as the five **A**s:

1. **A**dmissibility must guide actions: document everything that is done;

2. **A**cquire the evidence without altering or damaging the original;

3. **A**uthenticate your copy to be certain it is identical to the source data;

4. **A**nalyze the data while retaining its integrity; and,

5. **A**nticipate the unexpected.

These cardinal rules are designed to facilitate a forensically sound examination of computer media and enable a forensic examiner to testify in court as to their handling of a particular piece of evidence. A forensically sound examination is conducted under controlled conditions, such that it is fully documented, replicable and verifiable. A forensically sound methodology changes no data on the original evidence, preserving it in pristine condition. The results must be replicable such that any qualified expert who completes an examination of the media employing the same tools and methods employed will secure the same results.

After reading this paper, you may know enough of the basics of computer forensics to conduct a rudimentary investigation; but recognize that conducting a computer forensic investigation without the assistance of a qualified expert is a terrible idea. Experiment on an old system if you'd like, but leave real evidence to the experts.

Computer forensics focuses on three categories of data:

Active Data: These are the current files on the computer, still visible in directories and available to applications. Active data may be readily comprehensible using simple translation techniques (i.e., plain text files), but will more often need to be viewed within an application (computer program) to be useful. Such applications range from e-mail clients like Outlook, to database programs like Access or Excel, to word processors like Word or WordPerfect. Active data may also be password protected or encrypted, requiring further forensic activity to be accessed. Active data includes system data residing within the recycle bin, history files, temporary Internet directory, cookie "jar," system registry files, logs and other obscure but oft-revealing data caches. One important evidentiary point about data on a hard drive is that no matter what it may represent, whether simple text or convoluted spreadsheets, it exists only as infinitesimal magnetic flux reversals representing ones and zeroes *which must be processed by software to be intelligible*. Put another way, only the physical level with the magnetic domains is real; this level is also the least accessible. Words, pages, files, and directories are abstractions—illusions if you prefer--created by software that may or may not be reliable. The more levels of abstraction, the more likely evidence will not be, and should not be, admitted without scrutiny.

Latent Data: Latent data (also called "**ambient data**") are deleted files and other data, including memory "dumps" that have "lodged in the digital cracks" but can still be retrieved. This data resides on the hard drive or other storage media in, e.g., unallocated clusters (areas marked available for data storage but not yet overwritten by other data) and slack space. Latent data also includes information not readily understood absent special techniques and tools, like swap files, temporary files, printer spool files, metadata and shadow data (all discussed herein). The recovery of latent data is the art most often associated with computer forensics, but the identification, preservation, interpretation and management of active data is no less demanding of a forensic expert's skill.

Archival Data: This is data that's been transferred or backed up to peripheral media, like tapes, CDs, DVDs, ZIP disks, floppy disks, network servers or the Internet. Archival data can be staggeringly voluminous, particularly in a large organization employing frequent, regular back up procedures. It is critically important to recognize that an archival record of a source media never reflects all of the data that can be identified and extracted from the source media because such back ups don't carry forward latent data. Accordingly, an opponent's offer to furnish copies of back up tapes is, while valuable, no substitute for a forensic examination of a true bit-by-bit copy of the source disk drive.

Tell It to the Judge

Imagine that a case comes in where the content of a personal computer is critically important. Perhaps your client's marriage is on the rocks and infidelity and hidden assets are at issue. If you represent the wife, do you think that the philandering husband is going to agree to make his personal computer available to you; handing over the chat room transcripts, cyber-sex sessions, incriminating e-mails, Quicken balances, Internet history files, brokerage account records, digital photographs of the fluff on the side, business trip expense records, overseas account passwords and business correspondence? Chances are Hubby is going to fight you tooth and nail and, when finally ordered to make the machine available, he will clumsily seek to delete anything deemed compromising. But even if Hubby isn't trying to cover his tracks, know that every time he saves a file, or starts a program—*in fact every time he simply boots the machine*—some latent data is altered or overwritten to the point it can never be retrieved. By way of example, Windows accesses (and thus modifies metadata for) about a thousand files every time it boots up (and you wondered why booting took so long)!

"And just what was that little window you clicked off when I came in?"

You must persuade the court that conventional paper discovery is inadequate and that your client's interests will be irreparably harmed if she isn't granted access to Hubby's computer and afforded the right to conduct a complete forensic examination of same, starting with the creation of a sector-by-sector bit stream copy of the hard drive. Because Hubby has hired a savvy advocate, the judge is being assured that all reasonable steps have been taken to identify and protect computer data and that print outs of discoverable material will be furnished, subject to claims of privilege and other objections. If you can't articulate why your opponent's proposal is hogwash and thoroughly educate the judge about the existence and ongoing destruction of latent data, Missus is out-of-luck.

To be prepared to educate the Court, evaluate and select a computer forensics effort or simply better understand and advise your clients about "safe" data practices, you need a working knowledge of how a computer stores data and, more to the point, where and how data lives on after it's supposed to be gone.

To get that working knowledge, this section explains (as simply and painlessly as possible) the nuts and bolts of computer storage, beginning with the bits and bytes that are the argot of all digital computing, then on to the mechanics of hard drive operation and finally to the nooks and crannies where data hides when it doesn't want to be dispatched to that big CPU in the sky.

Bits and Bytes

You can become very facile with computers never knowing the nitty-gritty about bits and bytes, but when it comes to building a fundamental understanding of computer forensics, you've got to

begin with the building blocks of computer data: bits and bytes. You know something of bits and bytes because every computer ad you've seen uses them in some impressive-sounding way. The capacity of computer memory (RAM), size of computer storage (disks), and the data throughput speed of modems and networks are all customarily expressed in bits and bytes.

This Little Piggy went to Market

When we express a number like 9,465 in the decimal system, we understand that each digit represents some decimal multiple. The nine is in the thousands place, the four in the hundreds, the six in the tens place and so on. You could express 9,465 as: *(9 x 1000) + (4 x 100) + (6 x 10) + (5 x 1),* but check writing would quickly become an even more tedious chore. We just know that it is a decimal system and process the string 9,465 as nine thousand four hundred sixty-five.

Another equivalent method would be to use powers of ten. We can express 9,645 as: $(9 \times 10^3) + (4 \times 10^2) + (6 \times 10^1) + (5 \times 10^0)$. This is a "base-ten" system.

We probably came to use base ten in our daily lives because we evolved with ten fingers and ten toes, but had we slithered from the primordial ooze with eight or twelve digits, we could have gotten along quite nicely using a base-eight or base-twelve system. The point is that any number and consequently any datum can be expressed using any number system, and computers use the "base-two" or binary system.

A Bit about the Bit

Computers use binary numbers, and therefore **binary digits** in place of decimal digits. The word **bit** is even a shortening of the words "Binary digIT." Unlike the decimal system, where any number is represented by some combination of ten possible digits (0-9), the bit has only two possible values: zero or one. This is not as limiting as one might expect when you consider that a digital circuit—essentially an unfathomably complex array of switches—hasn't got ten fingers to count on, but is very, very good and darn fast at being "on" or "off." In the binary system, each binary digit—"bit"—holds the value of a power of two. Therefore, a binary number is composed of only zeroes and ones, like this: 10101. How do you figure out what the value of the binary number 10101 is? You do it in the same way we did it above for 9,465, but you use a base of 2 instead of a base of 10. Hence: $(1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 16 + 0 + 4 + 0 + 1 = 21$.

As you see, each bit holds the value of increasing powers of 2, standing in for zero, two, four, eight, sixteen, thirty-two, sixty-four and so on. That makes counting in binary pretty easy. Starting at zero and going through 21, decimal and binary equivalents look like the table at right.

0 = 0	11 = 1011
1 = 1	12 = 1100
2 = 10	13 = 1101
3 = 11	14 = 1110
4 = 100	15 = 1111
5 = 101	16 = 10000
6 = 110	17 = 10001
7 = 111	18 = 10010
8 = 1000	19 = 10011
9 = 1001	20 = 10100
10 = 1010	21 = 10101

Still unsure why this is important forensically? Hang in there!

I'll Byte

The simplest definition of a byte is that it is a string of eight bits, perhaps 10011001 or 01010101 or 11111111 or any other eight digit binary variation. The biggest number that can be stored as one byte of information is 11111111, equal to 255 in the decimal system. The smallest number is zero or 00000000. Thus, there are only 256 different numbers that can be stored as one byte of information. Any number that is greater than 255 has more than eight bits when written out in binary, and needs at least two bytes to be expressed.

"When you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the state of science."
- Lord Kelvin

Computers need to work with words as well as numbers, so how do we get from numbers to letters of the alphabet? Computers use a coded set of numbers to represent letters, both upper and lower case, as well as punctuation marks and special characters. This set of numbers is known as the **ASCII** code (for **American Standard Code for Information Interchange,** pronounced "ask-key"), and is commonly used by many different types of computers. By limiting the ASCII character set to less than 256 variations, each letter (or punctuation mark) can be stored as one byte of information in the computer's memory. A byte can also hold a string of bits to express other information, such as the description of a visual image, like the pixels or colors in a photograph. The byte, then, is the basic unit of computer data.

Why is an eight-bit string the fundamental building block of computing? It just sort of happened that way. In this time of cheap memory, expansive storage and lightning-fast processors, it's easy to forget how very scarce and costly all these resources were at the dawn of the computing era. Eight bits was basically the smallest block of data that would suffice to represent the minimum complement of alphabetic characters, decimal digits, punctuation and special instructions desired by the pioneers in computer engineering. It was in another sense about all the data early processors could chew on at a time, perhaps explaining the name "byte" coined by IBM.

"01101001, 00111011, 00011010, but, but!"

Storing alphanumeric data one character to a byte works in places that employ a twenty-six letter alphabet, but what about countries like China, Japan or Korea where the "alphabet" consists of thousands of characters? To address this, many applications dedicate two bytes to recording each character, with the most widely accepted double-byte character system called **Unicode**.

Now it may seem that you've asked for the time and been told the history of clock making, but computer forensics is all about recorded data, and all computer data exists as bits and bytes. What's

more, you can't tear open a computer's hard drive and find tiny strings of ones and zeros written on the disk, let alone words and pictures. The billions of bits and bytes on the hard drive exist only as faint vestiges of magnetism, microscopic in size and entirely invisible. It's down here--way, way down where a dust mote is the size of Everest and a human hair looks like a giant sequoia--where all the fun begins.

Information Storage

We store information by translating it into a physical manifestation: cave drawings, Gutenberg bibles, musical notes, Braille dots or undulating grooves in a phonograph record. Because binary data is no more than a long, long sequence of ones and zeros, it can be recorded by any number of alternate physical phenomena. You could build a computer that stored data as a row of beads (the abacus), holes punched in paper (a piano roll), black and white vertical lines (bar codes) or bottles of beer on the wall (still waiting for this one!).

"I should have had him put into a more manageable format years ago."

But if we build our computer to store data using bottles of beer on the wall, we'd better be plenty thirsty because we will need something like 99,999,999 bottles of beer to get up and running. And we will need a whole lot of time to set those bottles up, count them and replace them as data changes. Oh, and we will need something like the Great Wall of China to set them on. Needless to say, despite the impressive efforts ongoing at major universities and bowling alleys to assemble the raw materials, our beer bottle data storage system isn't very practical. Instead, we need something compact, lightweight and efficient—a leading edge technology--in short, a refrigerator magnet.

Magnetic Storage

Okay, maybe not a refrigerator magnet exactly, but the principles are the same. If you take a magnet off your refrigerator and rub it a few times against a metal paperclip, you will transfer some magnetic properties to the paperclip. Suppose you lined up about a zillion paper clips and magnetized some but not others. You could go down the row with a piece of ferrous metal (or, better yet, a compass) and distinguish the magnetized clips from the non-magnetized clips. Chances are this can be done with less space and energy than beer bottles, and if you call the magnetized clips "ones" and the non-magnetized clips "zeroes," you've got yourself a system that can record binary data. Were you to glue all those paper clips in concentric circles onto a spinning phonograph record and substitute an electromagnet for the refrigerator magnet, you wouldn't be too far afield of what goes on inside the hard and floppy disk drives of a computer, albeit at a much smaller scale. In case you wondered, this is also how we record sounds on magnetic tape, except that instead of just determining that a spot on the tape is magnetized or not as it rolls by, we gauge varying degrees of magnetism which corresponding to variations in the recorded sounds. This is called **analog** recording—the variations in the recording are analogous to the variations in the music.

Since computers process electrical signals much more effectively than magnetized paper clips jumping onto a knife blade, what is needed is a device that transforms magnetic signals to electrical signals and vice-versa—an energy converter. Inside every floppy and hard disk drive is a gadget called a disk head or read/write head. The read/write heads are in essence tiny electromagnets that perform this conversion from electrical information to magnetic and back again. Each bit of data is written to the disk using an encoding method that translates zeros and ones into patterns of magnetic flux reversals. Don't be put off by Star Wars lingo like "magnetic flux reversal"--it just means flipping the magnet around to the other side or "pole."

Older hard disk heads make use of the two main principles of electromagnetic force. The first is that applying an electrical current through a coil produces a magnetic field; this is used when writing to the disk. The direction of the magnetic field produced depends on the direction that the current is flowing through the coil. The converse principle is that applying a magnetic field to a coil will cause an electrical current to flow. This is used when reading back previously written information. Newer disk heads use different physics and are more efficient, but the basic approach hasn't changed: electricity to magnetism and magnetism to electricity.

Fantastic Voyage

Other than computer chip fabrication, there's probably no technology that has moved forward as rapidly or with such stunning success as the hard disk drive. Increases in capacity and reliability, precision tolerances and reduction in cost per megabyte all defy description without superlatives. The same changes account for the emergence of electronic media as the predominant medium for information storage (it's big—it's cheap—it's reliable), with commensurate implications and complications for civil discovery.

Since you now understand the form of the information being stored and something of the physical principles underlying that storage, it's time to get inside the hard drive and draw closer to appreciating where and why data can be deleted but still hang around. In 1966, Hollywood gave us the movie "Fantastic Voyage" about a group of scientists in a submarine shrunken down to microscopic dimensions and injected into the bloodstream. A generation later, the Magic School Bus made similar journeys. Let's do likewise and descend deep within the inner workings of a hard drive.

Caveat: At this point, we start talking about the innards of a personal computer. Should you be tempted to actually open one up and monkey around inside, please be advised that there is a significant risk of damage to the computer, your data and, most importantly, *to you*. Before you open the case of any PC, pull the plug and disconnect all cables, especially the power, modem, monitor and printer cables. Resist all temptation to poke around inside the power supply. There's little worth seeing in there and you can electrocute yourself. Seriously! If you experiment on a hard drive, be sure it contains no data that you care to retain. Note also that the technical term for a hard drive that has been opened up is "toast."

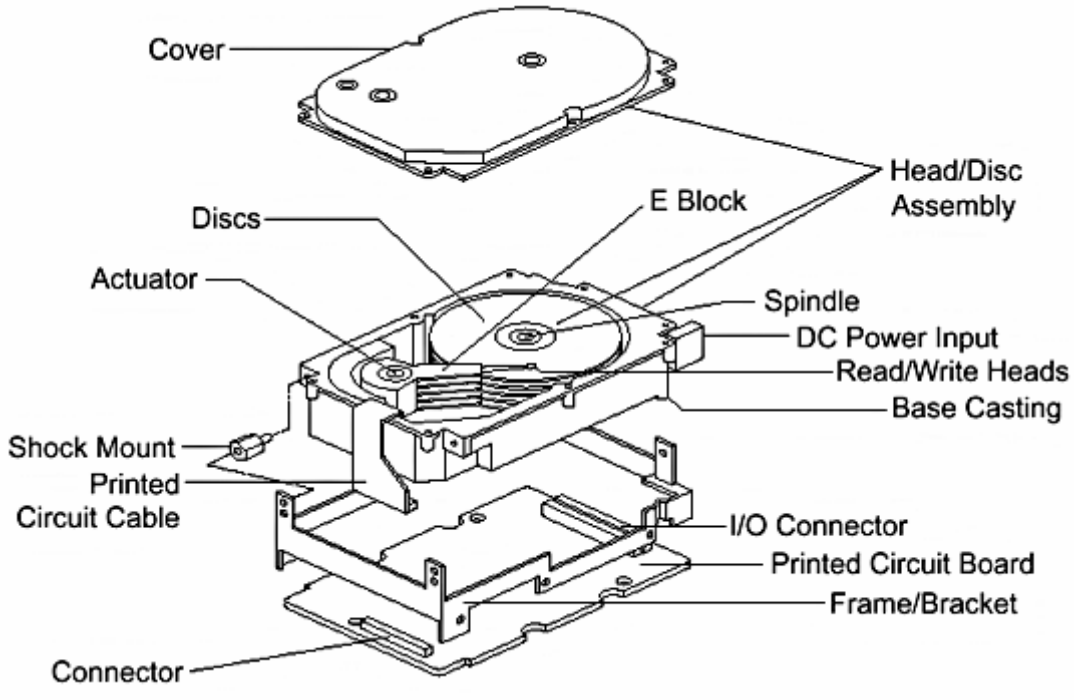
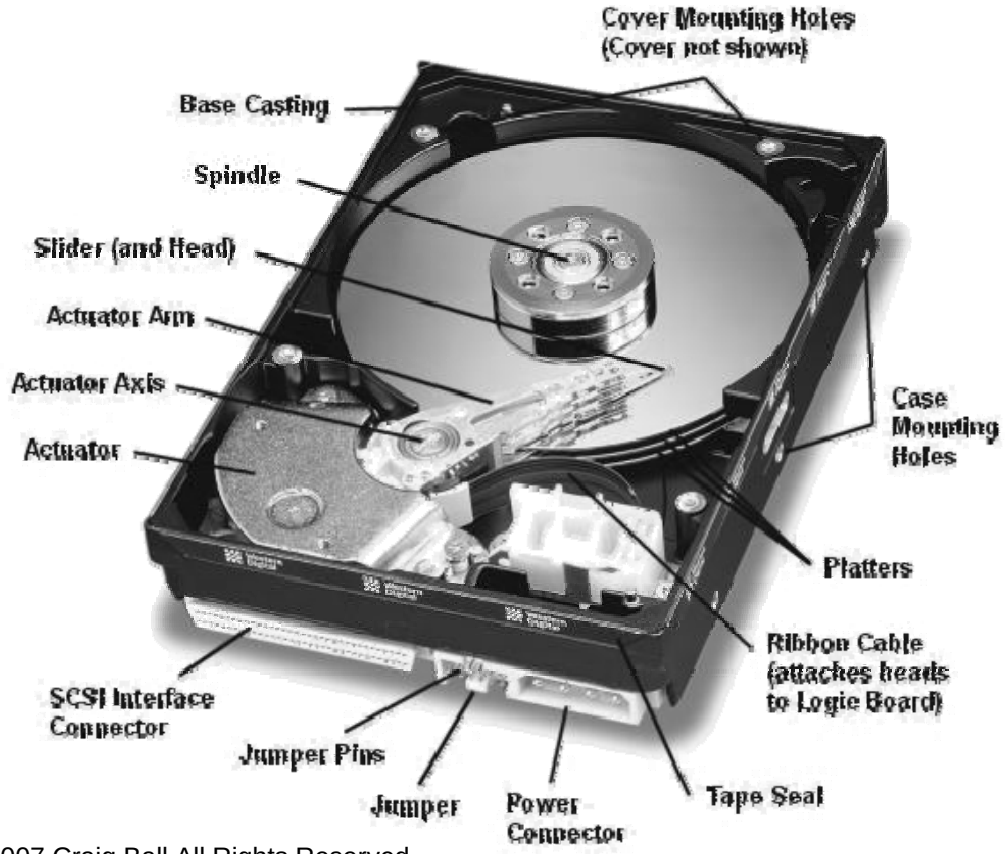


Figure 1

(Above) This is an exploded view of a typical personal computer hard drive. Note the stack of discs (platters) and the ganged read/write heads. (Below) A photo of a hard drive's interior with cover removed.



Disc Anatomy 101

A personal computer hard drive is a sealed aluminum box measuring (for a desktop system) roughly 4" x 6" x 1" in height. Though often mounted above or below the optical (CD/DVD) drives, it is not uncommon to encounter the hard drive located almost anywhere within the case, customarily secured by several screws attached to any of six or more pre-threaded mounting holes along the edges of the case. One face of the case will be labeled to reflect the drive specifications as in Fig. 2, while a printed circuit board containing logic and controller circuits will cover the opposite face (shown removed in Fig. 3).

Figure 2

Hard disk drives principally use one of three common interfaces: IDE/ATA, SCSI and S-ATA. You can tell immediately by looking at the back of the hard disk which interface is being used by the drive:

- IDE/ATA (Parallel ATA): A 40-pin rectangular connector (Figs. 4 and 5).
- SCSI: A 50-pin, 68-pin, or 80-pin D-shaped connector (see fig. 1).
- S-ATA (serial ATA): A 7-pin flat connector, less than a third the size of its IDE counterpart (Fig 5)

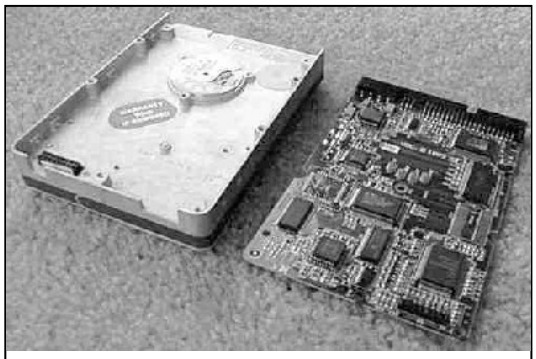


Figure 3

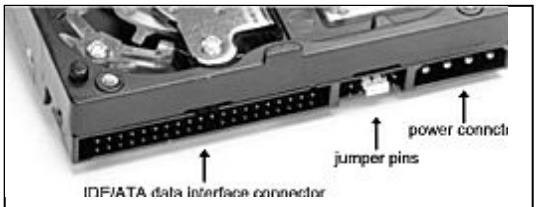


Figure 4

A hard disk contains round, flat discs called **platters**, coated on both sides with a special material able to store data as magnetic patterns. Much like a record player, the platters have a hole in the center allowing them to be stacked on a spindle. The platters rotate at high speed—typically 5,400, 7,200 or 10,000 rotations per minute--driven by a special motor. The read/write heads are mounted onto sliders and used to write data to the disk or read data from it. The sliders are, in turn, attached to arms, all of which are joined as a single assembly oddly reminiscent of a record player's tone arm and steered across the surface of the disk by a

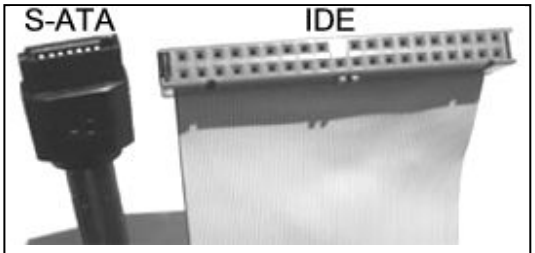
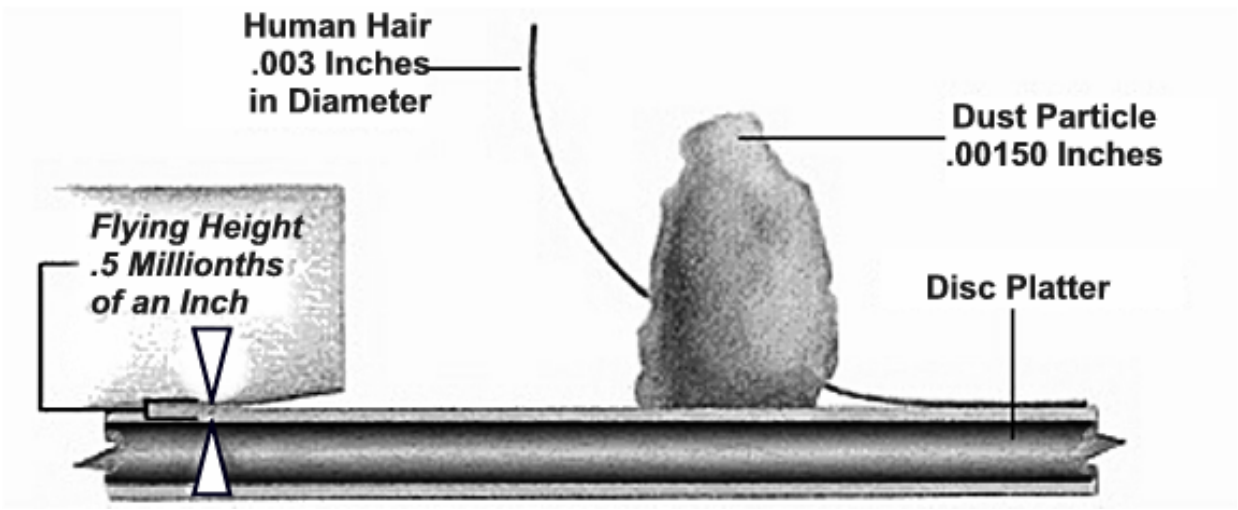


Figure 5

device called an actuator. (Fig. 6). Each platter has two heads, one on the top of the platter and one on the bottom, so a hard disk with three platters (normally) has six surfaces and six total heads.

Figure 6

When the discs spin up to operating speed, the rapid rotation causes air to flow under the sliders and lift them off the surface of the disk--the same principle of lift that operates on aircraft wings and enables them to fly. The head then reads the flux patterns on the disc while flying just *.5 millionths of an inch* above the surface. At this speed, if the head bounces against the surface, there is a good chance that the heads or sliders would burrow into the media, obliterating data and frequently rendering the hard drive inoperable ("head crash"). Surprisingly, head crashes are increasingly rare events even as the tolerances have become more exacting. To appreciate the fantastic tolerances required for achieving this miracle, consider Fig. 7. A human hair is some *6,000 times thicker* than the flying height of a modern hard drive read/write head! No wonder hard drives must be assembled in "clean rooms" with specially filtered air supplies.



Perspective: Woody Monroy, head of corporate communications for hard drive maker Seagate Technology, L.L.C., points out that, in terms of speed and tolerances, a hard drive's operation is equivalent to *an F-16 jet fighter plane flying at 813 times the speed of sound and one-sixty second of an inch off the ground…while counting every blade of grass as it goes!*

Sectors, and Clusters and Tracks, Oh My!

Now it starts to get a little complicated, but stay with me because we've nearly unraveled the mystery of latent data. At the factory, platters are organized into specific structures to enable the organized storage and retrieval of data. This is **low level formatting,** dividing each platter into tens of thousands of densely packed concentric circles called **tracks**. If you could see them (and you can't because they are nothing more than microscopic magnetic traces), they might resemble the growth rings of the world's oldest tree. It's tempting to compare platter tracks to a phonograph record, but you can't because a phonograph record's track is a single spiraling groove, not concentric circles. A track holds far too much information to serve as the smallest unit of storage on a disk, so each one is further broken down into **sectors**. A sector is normally the smallest individually addressable unit of information stored on a hard disk, and holds **512 bytes** of information. The first PC hard disks typically held 17 sectors per track. Figure 8 shows a very simplified representation of a platter divided into tracks and sectors. In reality, the number of tracks and sectors is far, far greater. Additionally, the layout of sectors is no longer symmetrical, to allow the inclusion of more sectors per track as the tracks enlarge away from the spindle. Today's hard disks can have thousands of sectors in a single track and make use of a space allocation technique called zoned recording to allow more sectors on the larger outer tracks of the disk than on the smaller tracks nearer the spindle.

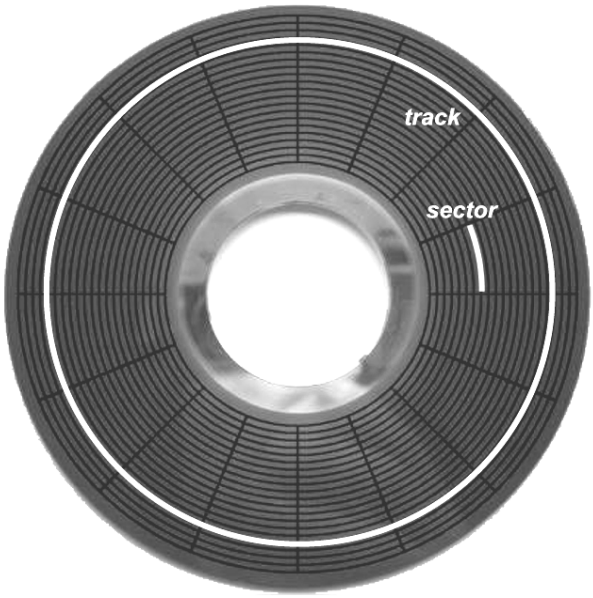


Figure 8

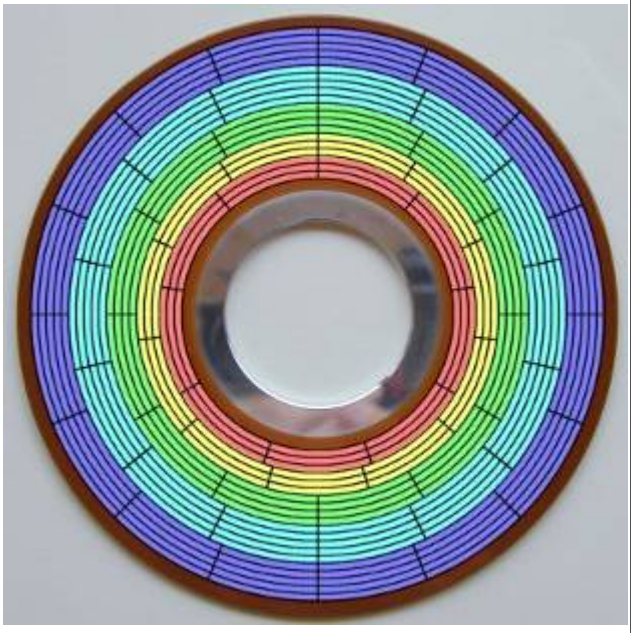


Figure 9

Figure 9 is an illustration of zoned recording. This model hard disk has 20 tracks divided into five zones, each shown as a different color (or shade of gray, if not printed in color). The outermost zone has 5 tracks of 16 sectors; followed by 5 tracks of 14 sectors, 4 tracks of 12 sectors, 3 tracks of 11 sectors, and 3 tracks of 9 sectors. Note that the size (length) of a sector remains fairly constant over the entire surface of the disk, unlike the non-zoned disk representation in Fig 8. Absent zoned recording, if the innermost zone were nine sectors, every track on this hard disk would be limited to only 9 sectors, greatly reducing capacity. Again, this is just an illustration; drives actually have thousands of tracks and sectors.

To this point, we have described only *physical*

units of storage. That is, platters, tracks, sectors and even bits and bytes exist as discrete *physical* manifestations written to the media. If you erase or overwrite data at the physical level, it's pretty much gone forever. It's fortunate, indeed, for forensic investigators, that personal computers manage data not physically but *logically*. Because it would be impractical to gather the megabytes of data that comprise most programs by assembling it from 512 byte sectors, the PC's operating system speeds up the process by grouping sectors into continuous chunks of data called **clusters**.

A cluster is the smallest amount of disk space that can be allocated to hold a file. Windows and DOS organize hard disks based on clusters, which consist of one or more contiguous sectors. The smaller the cluster size, the more efficiently a disk stores information. A cluster is also called an **allocation unit**.

The Numbers DO Lie

Hard drive specifications typically reference numbers of cylinders, sectors and heads. At one time, these numbers corresponded to genuine physical characteristics of the hard drive. Cylinders were the tracks on the platter, sectors were segments of cylinders of those cylinders and heads stated the actual number of read/write heads inside the case. When these were "real" numbers, you could use them to calculate the storage capacity of the drive. The most important thing to realize about these numbers today is that they are fictions and no longer have anything to do with what actually goes on inside the hard drive. This is a classic example of one branch of technology outstripping another and the workarounds needed to adapt to outdated standards. For years, the basic input output system (BIOS) of personal computers could only address a maximum of 1024 tracks, 16 heads and 63 sectors (540 MB), but the hard drive industry quickly moved far beyond those limitations. Consequently, the logic boards on modern hard drives must either manipulate the data stream to mimic the structure of older devices or, more commonly, have abandoned the obsolete cylinder/head/sector (CHS) addressing system in favor of what is called Logical Block Addressing (LBA).

Operating Systems and File Systems

Having finally gotten to clusters, the temptation to jump right into latent data is almost irresistible, but it's important that we take a moment to get up to speed with the DOS and Windows operating systems, and their file systems, or at least pick up a smattering of the lingo surrounding same so you won't be bamboozled deposing the opposition's expert.

As hard disks have grown exponentially in size, using them efficiently is increasingly more difficult. A library with thirty books runs much differently than one with 30 million. The **file system** is the name given to the logical structures and software routines used to control access to the storage on a hard disk system and the overall structure in which files are named, stored and organized. An **operating system** is a large and complex collection of functions, including the user interface and control of peripherals like printers. Operating systems build on file systems. If the operating system is the car, then the file system is its engine. Operating systems are known by familiar household names, like **MS-DOS**, **Windows or Vista**. In contrast, file systems go by obscure (and unflattering) monikers like **FAT, FAT32, VFAT** and **NFTS**. Rarely in day-to-day computer use must we be concerned with the file system, but it plays a critical role in computer forensics because the file system determines the logical structure of the hard drive, including its cluster size. The file system also determines what happens to data when the user deletes a file or subdirectory.

The FAT and NTFS File Systems

To simplify a complex subject, this topic will focus on the two file systems used in the Windows environment:, being the **FAT family of file systems** used by DOS, Windows 95-98 and Windows ME, as well as the **NTFS file system** at the heart of Windows NT, 2000, XP and Vista. Be advised that, although these file systems account for the vast majority (90+%) of personal computers in the world, there are non-Microsoft operating systems out there, such as Unix, Linux, Apple, OS/2 and BeOS. Though similarities abound, these other operating systems use different file systems, and the Unix or Linux operating systems often lie at the heart of corporate and web file servers—today's "big iron" systems--making them increasingly important forensically. Perhaps not today, but within a few years, chances are you'll be seeking discovery of data residing on a machine running a flavor of Linux.

"Do you realize we've been together through three versions of Windows?"

The FAT Family

The FAT family refers not to the epidemic of obesity in America but to a lineage of file systems that organize the major disk structures of the hard drive, including **FAT12, FAT16, VFAT** and **FAT32**. FAT is short for **File Allocation Table**, referring to the table of contents that serves as a road map and card catalogue of every bit of data on the drive. The numbers refer to the number of bits used to label the clusters. Since more bits equals a longer address number and a longer address number equals the ability to store more clusters, using 2^{16} bits allowed the cataloguing of 65,536 clusters versus the parsimonious 4,096 clusters (2^{12}) permitted by a twelve bit cluster number.

As with so many aspects of the personal computer, the file system has undergone an evolutionary process spurred by limitations that didn't seem much like limitations at the time each system was designed. For example, the MS-DOS/Windows 3.X file system, known simply as FAT (and also, over time, called FAT12 and FAT16) was originally designed to manage floppy disks (DOS was, after all, short for **D**isk **O**perating **S**ystem). Its greatest virtue was simplicity, but a lack of security, reliability and support for larger hard discs proved its Achilles' heel. Not even the most prescient among us could have anticipated personal computer users would have access to inexpensive one terabyte hard drives. It was simply inconceivable as little as ten years ago. Accordingly, the DOS and Windows 3.X file systems used so limited a cluster numbering system that they were unable to create a disk partition (volume) larger than two gigabytes, and then only if large clusters were used, wasting a lot of disk space (something we will return to later). This limitation lasted right up through the first version of Windows 95! The need to address larger and larger hard drives was a prime mover driving the evolution of the FAT file system.

NTFS

If you spent much time using Microsoft operating systems built on the FAT file system, you don't have to be told how quirky and unreliable the computing experience can be. By the early 1990s, as the networking of personal computers was increasingly common and hard drives were growing by leaps and bounds, the limitations of the FAT family of file systems were all too obvious, and those limitations were keeping Microsoft from selling its operating systems in the lucrative corporate arena. Microsoft realized that if it was going to gain a foothold in the world of networked computers, it would need to retool its operating system "from the ground up."

The New Technology File System (NTFS) was Microsoft's stab at a more reliable, secure and adaptable file system that would serve to meet the needs of business users. The new system offered greater protection against data loss, security features at both the user and file levels (limiting who can view and what can be viewed in the networked environment) and support for both long file names and gargantuan hard drives. The NTFS also makes more efficient use of those larger hard drives.

The NTFS file system is at the center of Windows NT, 2000, XP and Vista. Windows XP has been around since 2001 and Windows Vista is now the only entry-level operating system sold by Microsoft; consequently, virtually every PC entering the marketplace today uses the NTFS file system.

NTFS has had a significant impact upon computer forensics as a consequence of the more detailed information stored about system usage. NTFS uses a very powerful and fairly complex database to manage file storage. One unique aspect of NTFS that sets it apart from FAT is that, if a file is small enough in size (less than about 1,500 bytes), NTFS actually stores the file in the Master File Table to increase performance. Rather than moving the read/write heads to the beginning of the disk to read the Master File Table entry, and then to the middle or end of the disk to read the actual file, the heads simply move to the beginning of the disk, and read both at the same time. This can account for a considerable increase in speed when reading lots of small files. It also means that forensic examiners need to carefully analyze the contents of the Master File Table for revealing information. Lists of account numbers, passwords, e-mails and smoking gun memos tend to be small files.

To illustrate this critical difference a different way, if both FAT and NTFS were card catalogues at the library, FAT would direct you to books of all sizes out in the stacks, and NTFS would have all volumes small enough to fit tucked right into the card drawer.

Understanding the file system is key to appreciating why deleted data doesn't necessarily go away. It's the file system that marks a data cluster as deleted though it leaves the data on the drive. It's the file system that enables the creation of multiple partitions where data can be hidden from prying eyes. Finally, it's the file system that determines the size of a disk cluster with the attendant persistence of data within the slack space. Exactly what all this means will be clear shortly, so read on.

Formatting and Partitioning

There is a fair amount of confusion—even among experienced PC users—concerning formatting and partitioning of hard drives. Some of this confusion grows out of the way certain things were done in "the old days" of computing, i.e., fifteen years ago. Take something called "low level formatting." Once upon a time, a computer user adding a new hard drive had to low-level format, partition, and then high-level format the drive. Low level formatting was the initial "carving out" of the tracks and sectors on a pristine drive. Back when hard drives were pretty small, their data density modest and their platter geometries simple, low level formatting by a user was possible. Today, low level formatting is done at the factory and no user ever low-level formats a modern drive. Never. You couldn't do it if you tried; yet, you will hear veteran PC users talk about it still.

For Windows users, your new hard drive comes with its low level formatting set in stone. You need only be concerned about the disk's partitioning into **volumes,** which users customarily see as drive letters (e.g., C:, E:, F: and so on) and its high level formatting, which defines the logical structures on the partition and places at the start of the disk any necessary operating system files. For the majority of users, their computer comes with their hard drive partitioned as a single volume (universally called C:) and already high level formatted. Some users will find (or will cause) their hard drive to be partitioned into multiple volumes, each appearing to the user as if it were an independent disk drive. From the standpoint of computer forensics, perhaps the most important point to remember about FAT partitions is that they come in three different "flavors" called **primary, extended DOS** and **logical.** Additionally, the primary partition can be designated "**active"** and **"inactive.** Only one partition may be designated as active at any given time, and that partition is the one that boots the computer. The forensic significance is that inactive partitions are invisible to anyone using the computer, unless they know to look for them and how to find them. Inactive partitions, then, are a place where users with something to hide from prying eyes may choose to hide it. One simple way to find an inactive partition is to run the FDISK command if the system uses DOS or Windows 95/98/ME. If the system uses Windows Vista, XP, NT or Windows 2000 don't use FDISK. Instead, use Disk Management*,* an enhanced version of FDISK, but BE VERY CAREFUL! You can trash a hard drive in no time if you make a mistake with these utilities.

Cluster Size and Slack Space

By way of review, a computer's hard drive records data in bits, bytes and sectors, all physical units of storage established by the hard disk drive's internal geometry in much the same way as the size and number of drawers in a filing cabinet are fixed at the factory. Sticking with the file cabinet metaphor, bits and bytes are the letters and words that make up our documents.

Sectors (analogous to pages) are tiny segments of thousands of concentric rings of recorded data. A sector is 512 bytes, never more or less. A sector is the smallest individually addressable physical unit of information used by a computer. Computer hard drives can only "grab" data in sector-size chunks.

A common paper filing system uses labeled manila folders assembled into a "red rope file" or master file for a particular case, client or matter. A computer's file system stores information

on the hard drive in batches of sectors called clusters. Clusters are the computer's manila folders and, like their real-world counterparts, collectively form files.
These files are the same ones that you create when you type a document or build a spreadsheet.

In a Windows computer, cluster size is set by the operating system when it is installed on the hard drive. Typically, Windows 98/ME ME clusters are 32 KB, while Windows Vista/XP/NT clusters are 4 KBs. Remember that a cluster (also called an allocation unit) is the smallest unit of data storage in a file system. You might be wondering, "what about bits, bytes and sectors, aren't they smaller?" Certainly, but as discussed previously, in setting cluster size, the file system strikes a balance between storage efficiency and operating efficiency. The smaller the cluster, the more efficient the use of hard drive space; the larger the cluster, the easier it is to catalog and retrieve data.

This balance might be easier to understand if we suppose your office uses 500-page notebooks to store all documents. If you have just 10 pages to store, you must dedicate an entire notebook to the task. Once in use, you can add another 490 pages, until the notebook won't hold another sheet. For the 501st page and beyond, you have to use a second notebook. The difference between the capacity of the notebook and its contents is its "wasted" or "slack" space. Smaller notebooks would mean less slack, but you'd have to keep track of many more volumes.

In the physical realm, where the slack in the notebook holds empty air, slack space is merely inefficient. But on a hard drive, where magnetic data isn't erased until it's overwritten by new data, the slack space is far from empty. When Windows stores a file, it fills as many clusters as needed. Because a cluster is the smallest unit of storage, the amount of space a file occupies on a disk is "rounded up" to an integer multiple of the cluster size. If the file being stored is small, even just a few bytes, it will still "tie up" an entire cluster on the disc. The file can then grow in size without requiring further space allocation until it reaches the maximum size of a cluster, at which point the file system will allocate another full cluster for its use. For example, if a file system employs 32-kilobyte clusters, a file that is 96 kilobytes in size will fit perfectly into 3 clusters, but if that file were 97 kilobytes, then it would occupy four clusters, with 31 kilobytes idle. Except in the rare instance of a perfect fit, a portion of the final storage cluster will always be left unfilled with new data. This "wasted" space between the end of the file and the end of the last cluster is slack space (also variously called "file slack" or "drive slack," and it can significantly impact available storage (Fig. 10).

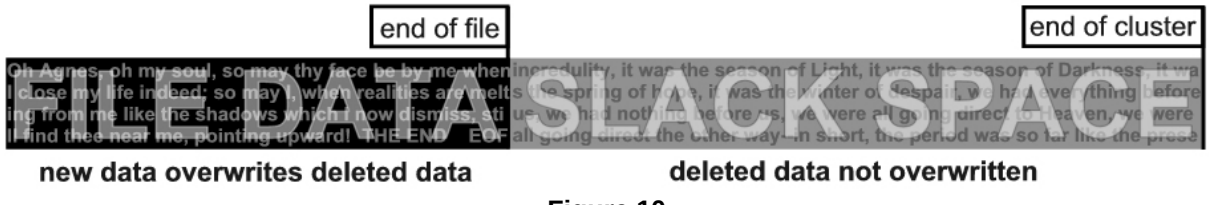


Figure 10

When Windows deletes a file, it simply earmarks clusters as available for re-use. When deleted clusters are recycled, they retain their contents until and unless the entire cluster is overwritten by new data. If later written data occupies less space than the deleted data, some of the deleted data remains, as illustrated in Figure 10. It's as if in our notebook example, when you reused notebooks, you could only remove an old page when you replaced it with a new one.

Though it might seem that slack space should be insignificant —after all, it's just the leftover space at the end of a file— the reality is that slack space adds up. If file sizes were truly random then, on average, one half of a cluster would be slack space for every file stored. But, most files are pretty small--if you don't believe it, take a look at your web browser's temporary Internet storage space. The more small files you have, the more slack space on your drive. It's not unusual for 25-40% of a drive to be lost to slack. Over time, as a computer is used and files deleted, clusters containing deleted data are re-used and file slack increasingly includes fragments of deleted files.

A simple experiment you can do to better understand clusters and slack space is to open Windows Notepad (usually in the Programs>Accessories directory). Type the word "hello" and save the file to your desktop as "hello.txt." Now, find the file you've just created, right click on it and select "properties." Your file should have a size of just 5 bytes, but the size it occupies on disk will be much larger, ranging from as little as 4,032 bytes in Windows XP or Vista to as much as 32,768 bytes in Windows 95 or 98. Now, open the file and change "hello" to "hello there," then save the file. Now, when you look at the file's properties, it has more than doubled in size to 11 bytes (the space between the words requires a byte too), but the storage space occupied on disk is unchanged because you haven't gone beyond the size of a single cluster

"True, I can't take it with me, but I can take the access codes to it."

Cluster size can vary depending upon the size of the hard drive volume and the version of FAT in use. The older versions of FAT which you encounter on computers using the first release of Windows 95 or any older version of Windows or DOS will create drives with cluster sizes ranging from 2,048 bytes (2K) to 32,768 bytes (32K). With the introduction of FAT32, introduced with Release 2 of Windows 95 and found in Windows 98, 2000, and ME cluster sizes have tended to be 32,768 bytes, particularly as hard drive size has ballooned. Under the NTFS file system found on Windows Vista, XP and NT, cluster size has dropped down to 4,032 bytes, resulting is less waste due to file slack.

Forensic Implications of Slack Space

In "Jurassic Park," scientists clone genetic material harvested from petrified mosquitoes to bring back the dinosaurs. Like insects in amber, Windows traps deleted data and computer forensics resurrects it. Though a computer rich with data trapped in file slack can yield a mother lode of revealing information, mining this digital gold entails tedious digging, specialized tools and lots of good fortune and patience.

The Windows system is blind to all information in the slack space. Searching is accomplished using a forensically-sound copy of the drive and specialized examination software, a hex editor utility that permits an examiner to read the data in each cluster directly from the media (or another operating system, like Linux, that treats a drive like a file), permitting string searches of contents. File slack is, by its very nature, fragmented, and the information identifying file type is the first data overwritten.

The search for plain text information is typically the most fruitful avenue in file slack examination and an exercise often measured not in hours, but in days or weeks of review. Experienced computer forensic examiners are skilled in formulating search strategies likely to turn up revealing data, but the process is greatly aided if the examiner has a sense of what he or she is seeking before the search begins. Are there names, key words or parts of words likely to be found within a smoking gun document? If the issue is trade secrets, are there search terms uniquely associated with the proprietary data? If the focus is pornography, is there image data or Web site address information uniquely associated with prohibited content?

Because most lawyers and litigants are unaware of its existence, file slack and its potential for disgorging revealing information is usually overlooked by those seeking and responding to discovery. In fairness, a request for production demanding "the contents of your computer's slack space" is unlikely to be productive. In practice, the hard drive must be examined by a computer forensics expert employed by one of the parties, a neutral expert agreed upon by all parties or a special master selected by the court.

Bear in mind that while the computer is running, computer data is constantly being overwritten by new data, creating a potential for spoliation. The most prudent course is to secure, either by agreement or court order, forensically-sound duplicates (clones or images) of potentially-relevant hard drives. Such specially created copies preserve both the live data and the information trapped in the slack space and other hiding places. Most importantly, they preserve the status-quo and afford litigants the ability to address issues of discoverability, confidentiality and privilege without fear that delay will result in destruction of data. There's more on this topic to follow.

How Windows Deletes a File

Increasingly, computer users have a vague awareness that when a file is deleted in Widows, it's not necessarily gone forever. In fact, Windows can be downright obstinate in its retention of data you don't want hanging around. Even actions like formatting a disk, long regarded as preemptive to data recovery, won't obliterate all your secrets—far from it (see "The BIG Lie"

sidebar, next page). Think about *that* next time you sell an old computer or donate it to the local high school!

How is that deleting a file doesn't, well, *delete* it? The answer lies in how Windows stores and catalogues files. Remember that the Windows files system deposits files at various locations on your disc drive and then keeps track of where it has tucked those files away in its File Allocation Table or Master File Table--essentially a table of contents for the massive tome of data on your drive. This table keeps tabs on what parts of the hard drive contain files and what parts are available for storing new data. When you delete a file, Windows doesn't scurry around the hard drive vacuuming up ones and zeroes. Instead, all it does is add a special hexadecimal character (E5h) to replace the first letter of the filename in FAT systems or add an entry to the master file table in NTFS that tells the system "this file has been deleted" and, by so doing, makes the disk space containing the deleted data available for storage of new data (called "**unallocated space**"). But deciding that a file drawer can be used for new stuff and clearing out the old stuff are two very different things. The old stuff—the deleted data—stays on the drive until it is magnetically overwritten by new data (and can even survive overwriting to some extent—but we're getting ahead of ourselves).

If we return to our library card catalogue analogy, pulling an index card out of the card catalogue doesn't remove the book from the shelves, though consulting the card catalog, alone, you'd think it's gone. Deleting a computer file only removes the card. The file (the "book" in our analogy) hangs around until the librarian needs the shelf space for new titles.

Let's assume there is a text file called secrets.txt on your computer and it contains the account numbers and access passwords to your Cayman Islands numbered account. Let's assume that the bloom has gone off the rose for you, marriage-wise, and you decide that maybe it would be best to get this file out of the house. So, you copy it to a thumb drive and then delete the original. Now, you're aware that though the file no longer appears in its folder, it's still accessible in the Recycle Bin. Consequently, you open the Recycle Bin and execute the "Empty Recycle Bin" command, thinking you can now rest easy. In fact, the file is not gone. All that has occurred is that Windows has flipped a bit in the Master File Table to signal that the space once occupied by the file is now available for reuse. The file,

The BIG Lie

Since the dawn of the personal computer, if you asked Microsoft, IBM, Compaq, Dell or others how to guard your privacy when selling or giving away a PC, chances are you'd be told to "delete the files and format your hard drive." If you followed this advice, DOS or Windows would solemnly warn you that formatting "will erase ALL data" on the disk." Trouble is, formatting doesn't erase all data. Not even close. This is the big lie. **Formatting *erases less than 1/10th of one percent* of the data on the disk**, such that anyone with rudimentary computer forensic skills can recover your private, privileged and confidential data. If it's not overwritten or physically destroyed, it's not gone. For a fine article on this issue, see the Jan/Feb 2003 issue of IEEE Security and Privacy Magazine or visit:
http://www.computer.org/security/garfinkel.pdf

and all of the passwords and account numbers it holds, is still on the drive and, until the physical space the data occupies is overwritten by new data, it's not that hard to read the contents of the old file or undelete it. Even if the file's overwritten, there's a chance that part of its contents can be read if the new file is smaller in size than the file it replaces. This is true for your text files, financial files, images, Internet pages you've visited and your e-mail.

If a computer has been in use for a while, odds are that it contains a substantial volume of unallocated file space and slack space containing "deleted" data. To illustrate, the old laptop computer on which this paper was originally written had 1.8 gigabytes of free space available on its 30--gigabyte hard drive, and *98.56% of that space contained deleted files*: 474,457 clusters of "deleted" data. How long that data remains retrievable depends on may factors, but one thing is certain: unless the computer user has gone to extraordinary lengths to eradicate every trace of the deleted data, bits and pieces--or even giant chunks of it--can be found if you know where and how to look for it.

What's this Hex Stuff, Voodoo?

Binary numbers get very confusing for mere human beings, so common shorthand for binary numbers is **hexadecimal notation**. If you recall the prior discussion of base-ten (decimal) and base-two (binary) notation, then it might be sufficient just to say that hexadecimal is base-sixteen. In hexadecimal notation, each digit can be any value from zero to fifteen. Accordingly, four binary digits can be replaced by just one hexadecimal digit and, more to the point; a byte can be expressed in just two hexadecimal digits. So 10110101 in binary is divided into two 4-bit pairs: 1011 and 0101. These taken individually are 11 and 5 in hexadecimal, so 10110101 in binary can be expressed as (11)5 in hexadecimal notation.

It's apparent that once you start using two digit numbers and parentheses in a shorthand, the efficiency is all but lost; but what can you do since we ten-fingered types only have 10 different symbols to represent our decimal numbers? Hexadecimal needs 16. The solution was to use the letters A through F to represent 10 through 15 (0 to 9 are of course represented by 0 to 9). So instead of saying (11)5, we say the decimal number 181 is "B5" in hexadecimal notation (or *hex* for short).

It's hard to tell if a number is decimal or hexadecimal just by looking at it: if you see "37", does that mean 37 ("37" in decimal) or 55 ("37" in hexadecimal)? To get around this problem, two common notations are used to indicate hexadecimal numbers. The first is the suffix of a lower-case "h". The second is the prefix of "0x". So "B5 in hexadecimal", "B5h" and "0xB5" all mean the same thing (as does the somewhat redundant "0xB5h"). Since a set of eight bits (two hexadecimal digits) is called a *byte,* the four bits of a single hexadecimal digit is called a "nybble" *(I'm not making this up!).*

The significance of hexadecimal notation in computer forensics goes beyond the use of hex byte E5h as a tag used in FAT to mark that the clusters occupied by a file as available for use, i.e., "deleted." Hexadecimal notation is also typically employed (alongside decimal and ASCII translations) in forensic software used for byte-by-byte and cluster-by-cluster examinations of hard drives.

RAM Slack

So far we've talked about recovering the remnants of files that a computer user purposefully stored and deleted. Suppose there were ways to gather bits and pieces of information the user deemed so secret he or she *never* knowingly stored it on the disk drive, perhaps a sensitive report read onscreen from floppy but not copied, a password or an online query. A now-defunct peculiarity in the DOS and earliest Windows file systems makes this possible, but the contents of the data retained are as unpredictable as a pull on a slot machine. These digital lagniappes reside in regions of the drive called **"RAM slack."**

To understand RAM slack, we need to review part of our discussion of file slack. Computers work with data in fixed block lengths called sectors and clusters. Like Nature, a computer abhors a vacuum, so sectors and clusters are always full of *something.* Earlier, we focused on file slack, the data that filled the space remaining when a file couldn't fill the last cluster of space allocated for its use, deleted data that remained behind for prying eyes to see. This data could range from as little as one byte to as much as 32,767 bytes of deleted material on a typical PC running Windows 98 (eight times less for Windows XP systems). This may not seem like much, but the entire text of the U.S. Constitution plus the Bill of Rights can be stored in less than 32,000 bytes!

Recall that file slack extends from the end of the file stored in the cluster until the end of the *cluster*, but what about the morsel of slack that exists between the end of the stored file and the end of the last *sector.* Remember that sectors are the smallest addressable unit of storage on a PC and are strung together to form clusters. Sectors are only 512 bytes in size and the computer, when it writes any data to disk, *will not write less than a full sector*. But what if the file data being written to the last sector can't fill 512 bytes and there is some slack remaining? If the sector has space remaining in its 512 bytes which it can't fill from the file being stored, older file systems padded the remaining space with whatever happened to be in the computer's Random Access Memory (RAM) at that moment, hence the name "RAM slack" (see Fig. 11). Granted, we are not talking about a whole lot of data—always less than 512 bytes—but it was enough for a password, encryption key, paragraph of text, or a name, address and phone number. Everything you do on a computer filters through the computers RAM, even if you don't save it to disk; consequently, *RAM slack can contain anything, and there are at least as many instances of RAM slack on a computer that has been in use for any length of time as there are files on the hard drive.*

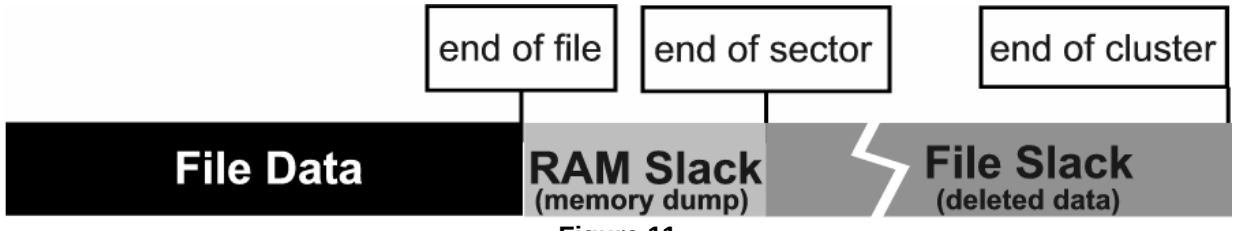


Figure 11

Swap Files

Just like you and me, Windows needs to write things down as it works to keep from exceeding its memory capacity. Windows extends its memory capacity (RAM) by swapping data to and from a particular file called a **"swap file**." When a multitasking system such as Windows has too much information to hold in memory at once, some of it is stored in the swap file until needed. If you've ever wondered why Windows seems to always be accessing the hard drive, sometimes thrashing away for an extended period, chances are it's reading or writing information to its swap file. Windows Vista, XP, NT and 2000 use the term **"page file"** (because the blocks of memory swapped around are called *pages),* but it's essentially the same thing: a giant digital "scratch pad."

Like RAM slack of yore, the swap file still contains data from the system memory; consequently, it can contain information that the typical user never anticipates would reside on the hard drive. Moreover, we are talking about a considerable volume of information. How much varies from system-to-system, but it runs to millions and millions of bytes. For example, the page file on the XP laptop used to write this article is currently about *1.6 gigabytes in size*. As to the contents of a swap file, it's pretty much a sizable swath of whatever kind of information exists (or used to exist) on a computer, running the gamut from word processing files, e-mail messages, Internet web pages, database entries, Quicken files, you name it. If the user used it, parts of it are probably floating around somewhere in the Windows swap file.

The Windows swap file sounds like a forensic treasure trove—and it is—but it's no picnic to examine. The data is usually in binary form—often without any corollary in plain text--and so must be painstakingly gone through, byte-by-tedious-byte. My 1.6 gigabyte page file might represent *sixteen million pages* of data. Although filtering software exists to help in locating, e.g., passwords, phone numbers, credit card numbers and fragments of English language text, it's still very much a needle-in-a-haystack effort (like so much of computer forensics in this day of vast hard drives).

Swap files have different names and may be either permanent or temporary on different versions of Windows. Users can adjust their system settings to vary the permanency, size or location of swap files. The table below lists the customary swap file name and location in each of the versions of Windows, but because these settings are user-configurable, there is no guarantee that the location will be the same on every system.

Because the memory swapping is (by default) managed dynamically in Windows 95, 98 and ME, the size of the swap file changes as needed, with the result that (barring custom settings by the user), the swap file in these versions tends to disappear each time the system is rebooted, its contents relegated to unallocated space and recoverable in the same manner as other deleted files.

Windows Version	Swap File Name	Typical Location(s)
Windows 3.1	386SPART.PAR	Root directory (C:\) Windows subdirectory Windows\System subdirectory
Windows 95, 98, ME	WIN386.SWP	Root directory (C:\)
Windows NT, 2000, XP, Vista	PAGEFILE.SYS	Root directory (C:\)

Windows NTFS Log File

The NTFS file system increases system reliability by maintaining a log of system activity. The log is designed to allow the system to undo prior actions if they have caused the system to become unstable. While arguably less important forensically in the civil setting than in a criminal matter, the log file is a means to reconstruct aspects of computer usage. The log file is customarily named $LogFile, but it is not viewable in Windows Explorer, so don't become frustrated looking for it.

TMP, BAK and Spool Files

Every time you run Microsoft Word or WordPerfect, these programs create temporary files containing your work. The goal of temp files is often to save your work in the event of a system failure and then disappear when they are no longer needed. In fact, temp file do a pretty good job saving your work but, much to the good fortune of the forensic investigator, they often do a pretty lousy job of disappearing. Temp files are often abandoned, frequently as a consequence of a program lock up, power interruption or other atypical shut down. When the application is restarted, it creates new temp file, but rarely does away with the predecessor file. It just hangs around indefinitely. Even when the application does delete the temp file, the contents of the file tend to remain in unallocated space until overwritten, as with any other deleted file.

As an experiment, search your hard drive for all files with the .TMP extension. You can usually do this with the search query "*.TMP." You may have to adjust your system settings to allow viewing of system and hidden files. When you get the list, forget any with a current date and look for .TMP files from prior days. Open those in Notepad or WordPad and you may be shocked to see how much of your work hangs around without your knowledge. Word processing applications are by no means the only types which keep (and abandon) temp files.

Files with the .BAK extensions (or a variant) usually represent timed back ups of work in progress maintained to protect a user in the even of a system crash or program lock up. Applications, in particular word processing software, create .BAK files at periodic intervals. These applications may also be configured to save earlier versions of documents that have been changed in a file with a .BAK extension. While .BAK files are supposed to be deleted by the system, they often linger on.

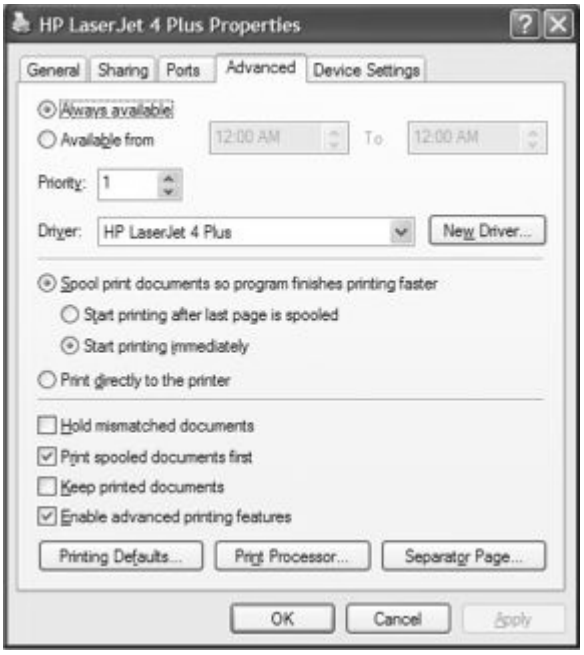


Figure 12

If you've ever poked around your printer settings, you probably came across an option for spooling print jobs, promising faster performance. See Figure 12 for what the setting box looks like in Windows XP. The default Windows setting is to spool print jobs so, unless you've turned it off, your work is spooling to the printer. Spool sounds like your print job is winding itself onto a reel for release to the print queue, but it actually is an acronym which stands for (depending upon who you ask) "simultaneous peripheral operations on line" or "system print operations off-line." The forensic significance of spool files is that, when spooling is enabled, anything you print gets sent to the hard drive first, with the document stored there as a graphical representation of your print job. Spool files are usually deleted by the system when the print job has completed successfully but here again, once data gets on the hard drive, we know how tenacious it can be. Like temp files, spool files occasionally get left behind for prying eyes when the program crashes. You can't read spool files as plain text. They must either be decoded (typically from either Windows enhanced metafile format or a page description language) or they must be ported to a printer compatible with the one for which the documents were formatted.

Windows Registry

The Windows Registry is the central database of Windows that stores the system configuration information, essentially every thing the operating system needs to "remember" to set it self up and manage hardware and software.

The registry can provide information of forensic value, including the identity of the computer's registered user, usage history data, program installation information, hardware information, file associations, serial numbers and some password data. The registry is also where you can access a list of recent websites visited and documents created, often even if the user has taken steps to delete those footprints. One benefit of the Registry in forensics is that it tracks the attachment of USB storage media like thumb drives and external hard drives, making it easier to track and prove data theft.

In a Windows 95/98/ME environment, the registry is a collective name for two files, USER.DAT and SYSTEM.DAT. In the Windows Vista/XP/NT/2000 environment, the registry is not structured in the same way, but the entire registry can be exported, explored or edited using a program called REGEDIT that runs from the command line (i.e., DOS prompt) and is found on all versions of Windows. You may wish to invoke the REGEDIT application on your system just to get a sense of the structure and Gordian complexity of the registry, but be

warned: since the registry is central to almost every function of the operating system, it should be explored with utmost care since its corruption can cause serious, i.e., *fatal, s*ystem errors.

Cookies

Cookies are the most maligned and misunderstood feature of web browsing. So much criticism has been heaped on cookies, I expect many users lump them together with computer viruses, spam and hacking as a Four Horseman of the Digital Apocalypse. Cookies are not malevolent; in fact, they enable a fair amount of convenience and function during web browsing. They can also be abused.

A cookie is a small (<4kb) text file that is deposited in a reserved cookie directory on a user's computer by a website visited by the user. It is, in a sense, a small scratch pad that can be used by a website to store information about the user so that the information can be retrieved by the website in a subsequent visit. Cookies are a means by which websites can personalize the user's online experience or speed the user's authentication. When you go to Amazon.com and the site greets you by name as soon as you arrive, such recognition occurs because the Amazon site has deposited a cookie on your machine during a prior visit. Cookies can contain many things, including a designated user name, a password you've created to access the site, a log of prior visits, customized settings and other data that allows the site to adapt to the user. Cookies can also record the address of the website a user visited just prior to arriving at the site depositing the cookie. When used to enhance and streamline a user's webs surfing, cookies are very beneficial to both user and website operator. It's important to note that cookies are not programs. They are merely electronic Post-It notes, but unscrupulous web site operators who, by working in concert, can assemble data about a user that will facilitate tracking a user's web surfing habits can abuse cookies.

From the standpoint of computer forensics, cookies offer insight to a user's online behavior. Users that take steps to erase their browser history files often forget to dispose of their cookies, which are stored in the cookies subdirectory of the Windows directory on Windows 95/98/ME systems and within the individual user profile on Windows Vista/XP/NT/2000 systems. On my system, I found 5,731 cookies. Very few of them represent any effort by me to customize anything on a website, but one that does is the cookie associated with my online subscription to the New York Times crossword puzzle, shown in Figure 13. Cookies are not required to adhere to any fixed format so note that very little of the cookie's content is intelligible. Most of the data has no value beyond the operation of the website that created it. However, note that the name of the cookie indicates (in Windows XP) the identity under which the user was logged in when the site was visited. The file's properties (not shown) will indicate the date the cookie was created and the date the web site was last accessed.

A file called INDEX.DAT contained within the Cookies subdirectory is worth examining since it contains a (partially) plain text listing of every site that dropped a cookie on the system, sort of a "super" history file. One provocative aspect of cookies is their ability to act as an authentication key. If the New York Times cookie from my system were copied to the Cookie subdirectory on your system, the New York Times website would see and admit you as me. This potential for extending an investigation using another person's cookie data raises many interesting—and potentially unsettling—possibilities.

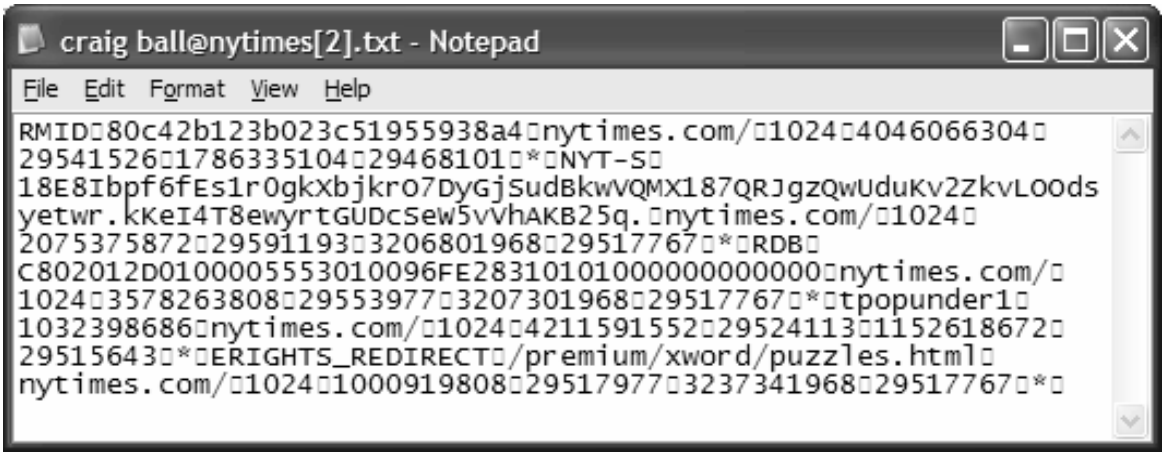


Figure 13

Application Metadata

Metadata is "data about data." *Application* metadata is a level of information embedded in a file and more-or-less invisibly maintained by the application that created the file. Although application metadata data security issues affect many programs, the epicenter of the application metadata controversy has been Microsoft Word and other components of Microsoft Office. Application metadata grows not out of the *Secret Bill Gates Conspiracy to Take Over the World*, but out of efforts to add useful features to documents, such as information on who created or edited a document, the document's usage and distribution history and much more. The problem with application metadata, especially for lawyers, comes about when people share Word document files. When you send someone (opposing counsel, a client, the court) a Word file on disk or via the Internet, you send not only the text and formatting of the document; you also transmit its application metadata layer. The associated metadata might reveal the amount of time spent editing the document and identify others with whom the document was shared. The metadata might also include collaborative commentary, earlier versions of the document and even the fact that you merely recycled a document prepared in another matter or purloined from another lawyer! In short, application metadata can cause problems ranging from embarrassment to malpractice.

In its Knowledge Base Article Q223396, Microsoft details some examples of metadata that may be stored in documents created in all versions of Word, Excel and PowerPoint, including:

- Your name
- Your initials
- Your company or organization name
- The name of your computer
- The name of the network server or hard disk where you saved the document
- Other file properties and summary information
- Non-visible portions of embedded OLE objects
- The names of previous document authors
- Document revisions
- Document versions
- Template information
- Hidden text
- Comments

While some application metadata is readily accessible just by viewing in the Office application, other application metadata can only be seen using a low-level binary file editor. Microsoft offers a free "Hidden and Collaboration Data Removal" utility for download. You can locate it by running a search at www.microsoft.com for "rhdtool.exe". While most metadata can be removed from Word documents, without buying any software, a simple and effective way to identify and eliminate metadata from Word documents is a $79.00 program called the Metadata Assistant, sold by Payne Consulting Group (www.payneconsulting.com).

Hidden Data

Most of what we have discussed thus far centers on data that no one has sought to conceal, other than by deletion. But, there are techniques by which data can be concealed on a computer, ranging from the unsophisticated and retrievable to the sophisticated and (practically) irretrievable. For example, files can be given the attribute "hidden" so as not to appear in directory listings. This is easily overcome by, e.g., issuing the dir /ah command, but you have to know to do this in your search. Data can be hidden in functional sectors marked as "bad" in the file table such that the systems simply skip these sectors. Here, the characterization of the sector will need to be changed or the sectors themselves will need to be examined to extract their contents. Earlier, this article discusses the use of inactive partitions to hide data; that is, hiding data in areas "unseen" by the operating system. Encrypted data poses near-insurmountable challenges if the encryption is sufficiently strong and unencrypted data hasn't found its way into swap files and slack space. Finally, and perhaps most insidious because of its simplicity, is the hiding of data in plain sight by simply changing its filename and file extension to seem to be something it is not, such as by renaming pornographic jpeg files as something that would not normally garner any attention, like "format.exe." Unless one compares file sizes or examines the files' contents and attributes with care, there would be little reason for a casual investigator to find the wolf in sheep's clothing.

Shadow Data

As previously discussed, data on a hard drive is stored in thousands of concentric rings called tracks over which a tiny read/write head flies, reading and writing information as

densely packed necklaces of magnetic fluctuations. This feat requires a mechanical precision unlike almost any other we encounter in our daily lives. But hard drives haven't always been as precise as modern disks and, in the days before mind-boggling data densities, minute variations in track alignment and in the size and penetration of the recording field were common. As a consequence, each time a track was overwritten, the read/write head might not completely cover the pre-existing data. Some of the magnetic information containing overwritten data may have "swerved" out of the track path due to wobbling in the head or other misalignment. Earlier disk writes may have occurred with the read/write head a bit further away from the surface of the disc, widening (and deepening) the bands of recorded data.

The consequence of this infinitesimal 3-D variation is that a remnant of previously recorded data can exist just beyond the borders of each track or at different levels in the physical media. This fringe of potentially recoverable information data is called **"shadow data."** Shadow data can potentially exist on older hard drives, floppy discs, backup tapes and Zip disks. Figure 14 is a graphical representation of what shadow data might look like on a disc drive if it were visible.

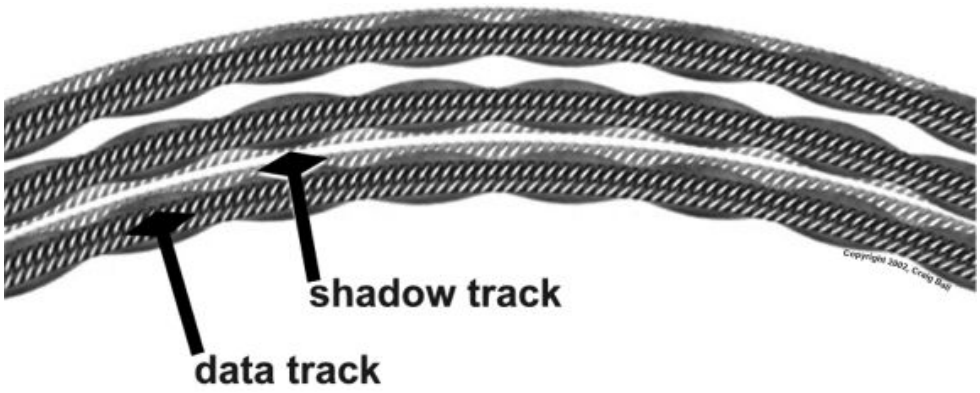


Figure 14

Shadow data is the DNA evidence of computer forensics, except that it's much, much harder and more costly to try to use shadow data and it faces significant admissibility hurdles. As a practical matter, shadow data still remains the exclusive province of shadowy three-letter government agencies and perhaps the "Q" branch of Her Majesty's Secret Service. Its extraction requires specialized equipment, and making any sense of it demands extraordinary tenacity and patience (not to mention a government-sized budget).

Other Revealing Data

In addition to the latent data possibilities described above, a thorough forensic investigation will look at a user's browser cache files (also called Temporary Internet Files in Internet Explorer), browser history files, web Bookmarks and Favorites and file dates. Of course, the user's e-mail and their Recycle Bin must also be explored. An alert investigator will also look

at the nature of software installed on a computer and the timing of that software's installation; that is, contextual analysis.

Contextual Analysis

The complexity and interactive nature of a personal computer permits revealing information to be gleaned not only from the contents of discrete files but also from the presence or absence of certain files and programs, as well as the timing of their appearance or disappearance. For example, the recent appearance of encryption or steganography applications (the latter employed to conceal data by invisibly integrating it within other carriers, usually drawings or photographs) may be a red flag that the user has hidden or encrypted data on the drive. The presence of a user-installed copy of the Quicken financial management program coupled with the absence of any financial data files may suggest that data has been removed from the machine. Similarly, the presence of a user-installed facsimile software program should trigger a search for facsimile image files.

If you were to examine usage patterns for a typical Windows PC, you'd find that more than 90% of the programs and files on the drive are never used in any given year. Most of us access the same little neighborhood of files and programs and rarely stray from them. This near-universal trait has both positive and negative implications for computer forensics. The positive is that the vignette of files likely to contain revealing information is small relative to the giant canvas of the hard drive, but the down side is that these needles hide in a very large haystack. If the discovery plan requires combing through or, worse, printing out "everything" on the drive, then it will be a gargantuan exercise, more than 90% wasted. If we focus instead only on those files that have been accessed or modified within a specified look back period, we need to have some basis on which to treat each file's date attributes as reliable. In fact, changing file dates is child's play and, absent an ability to validate the system clock at the time the attributes were applied, even dates that haven't been fudged may be fanciful without contextual analysis.

Going, Going, Gone

So far, this paper has spoken primarily of what information is available to find on a Windows personal computer and where it might be found. Now, we turn briefly to a few practical considerations in dealing with that data. If you look back at what we have covered heretofore, you'll see that a large volume of the potentially revealing information to be found is latent data, and the bulk of that data resides within unallocated space on the hard drive (e.g., in the slack space). Similarly, key forensic data like the swap files, TEMP files, log files and so forth are dynamic. They change constantly as programs are run and documents created. The point of all this is that unallocated space gets allocated and dynamic files change as a computer is used. For that matter, latent data can be progressively destroyed even when the computer is not in use, so long as the power is connected and the operating system is running. As hard as it is to obliterate *specific* data from a computer, some latent data is being completely destroyed all the time a computer is in operation, overwritten by new data. Your smoking gun is gradually being destroyed or, worse, may soon be disrupted by disk maintenance utilities that defragment the disk. Considering that Windows accesses and changes hundreds of files each time it boots, you can appreciate that doing nothing is

tantamount to allowing evidence to be destroyed. Every time you boot windows you destroy or alter data. The creation of temp files, the updating of logs, the reading of configuration files may all seem benign acts, but they likely entail use of unallocated space, overwriting of latent data and alteration of metadata values.

Bit Stream Backup

Once latent data is overwritten, it's pretty much gone forever. If you want to preserve the status quo and retain access to latent data, the only practical way to do so is by making a bit stream copy of the hard drive. A bit stream copy is a sector-by-sector/byte-by-byte copy of a hard drive. A bit stream copy preserves not only the files and directory structures; it preserves all of the latent data, too. Anything less will leave potential evidence behind. It's critically important that you appreciate the difference between a bit stream copy and an archival copy of the type that people create to protect them in the event of a system crash. Archival backups copy and retain only the active files on a drive, and frequently not even all of those. If you can imagine a hard drive with all latent data stripped away, you'd have a pretty good picture of an archival back up. In short, an archival back up is simply no substitute for a bit stream back up when it comes to computer forensics.

Computer forensic specialists create bit stream copies using any of several applications, including programs like Encase, FTK Imager, X-Ways Forensics, SnapBack and SafeBack. These and other commercially available programs make the mirroring process easier, but an identical copy of every sector of a drive can also be made using a free utility called Linux DD (which runs under the also-free Linux operating system, but not on a machine running DOS or Windows). Whatever program is used, it is essential that the examiner be able to establish its reliability and acceptance within the forensic community. The examiner should be able to demonstrate that he or she has a valid license to own and use the software as the use of a bootleg copy could prove an embarrassing revelation in cross-examination.

The creation of a forensically competent bit stream copy entails a second step. It is not enough to simply make a faithful copy of the disk drive; a forensic examiner must be equipped to irrefutably demonstrate that the copy does not deviate from the original, both immediately after it is created and following analysis. This is typically accomplished using some mathematical sleight-of-hand called **"hashing."** Hashing a disc creates a digital fingerprint; that is, a small piece of data that can be used to positively identify a much larger object. Hashing is a form of cryptography that relies upon a concept called "computational infeasibility" to fashion unique digital signatures. Essentially, the entire contents of any stream of digital information is processed by a specialized mathematical equation called an "algorithm" that works in only one direction because it would be a gargantuan (i.e., "computationally infeasible") task--demanding hundreds of computers and thousands of years--to reverse engineer the computation. The bottom line is that if the bit stream copy of the data is truly identical to the original, they will have the same hash values; but, if they differ by so much as a comma (well, a byte), the hash values will differ markedly. The computational infeasibility means that someone trying to pass a doctored drive off as a bit stream copy can't make changes that will generate an identical hash value. There are a number of hash algorithms floating around, but the two most frequently employed in

computer forensic work are called **MD5** and **SHA1**. Programs that create bit stream copies may also employ another form of authentication called **"Cyclic Redundancy Check" (CRC)**. CRC may be done before MD5 or SHA1 hashing or (less desirably) instead of it.

Computationally infeasible is not the same as computationally impossible, but it might as well be. From the standpoint of relative probabilities, before two hard drives with differing content could generate the same MD5 hash value ("hash collision"), you'd have won the lottery a *billion billion billion billion billion times*. That said, hash collisions have been contrived for the MD5 algorithm, but not in a manner that should give anyone pause in its near-term continued use to authenticate duplicate drives.

Now What?

But let's beam out of the digital domain and return to the practice of law on planet Earth. Either the opposition has computer data you want or you have computer data the other side may want. You now appreciate that evidence is potentially being destroyed as the computer is used. Now what?

When the government faces this dilemma, they have a pretty handy solution: get a warrant and seize everything. For the rest of us, getting, or even just preserving, computer evidence can be an uphill battle. If a computer is used to run a business, can you persuade the judge to order it be turned off and sequestered? If the computer is a mish-mash of personal, professional, private and privileged information, is it proper for the judge to order a wholesale copy of the hard drive to be turned over to the opposition? Where is the line between unwitting destruction of latent evidence and spoliation? These are not easy questions, but the law has generally recognized that the mere fact that the party opposing discovery has adopted a high tech filing system should not operate to deprive a party of access to discoverable material. If you would be entitled to inspect or copy the information were it on paper, why should that right be diminished because it's digitized?

When is Forensic Analysis Warranted?

"To a man with a hammer, everything looks like a nail," wrote Mark Twain. The same might be said of attorneys whose clients have benefited from the use of computer forensics in electronic discovery. Understandably, they want access to the other side's systems in every case. But, as powerful a tool as it is, computer forensic analysis probably has a place in less than one-in-ten litigated matters. The challenge for the court is identifying the issues and circumstances justifying forensic access, imposing appropriate safeguards and allocating the often-substantial cost.

It's long settled that evidence is discoverable whether it exists on paper or solely as a microscopic arc of magnetic data on a disk; but are we entitled to root around in another's computer hard drive when we couldn't do the same in their file room? The answer seems to be "occasionally." Absent a showing of abuse, the rules of procedure invest the responsibility to locate, preserve and produce discoverable material on the producing party. If the producing party responds "it's not there," the requesting party is largely bound to accept that representation unless there is some credible basis to suggest it's unreliable. But most people

lack the skill and tools to identify, preserve and extract latent computer data; so the statement "it's not there" is, at best, "it's not where we looked, and we haven't looked thoroughly."

By the same token, it's not reasonable to expect a responding party to hire a computer forensic examiner and perform a thorough search for latent data in every case. It's too expensive, time-consuming and not always certain to lead to the discovery of relevant evidence. Neither can the requesting party's forensic expert be granted unfettered access to an opponent's computers absent steps to protect the confidentiality of proprietary, privileged or just- downright-embarrassing material. A balance must be struck between the potential for discovery of relevant evidence and the potential for unwarranted intrusion at great expense.

The most obvious instance where forensic examination is indicated is a matter involving a credible allegation of negligent or intentional spoliation, or concealment, of electronic information or its paper counterpart. Another is a circumstance where it appears likely that relevant and discoverable data exists, but is accessible only through the use of forensic restoration techniques. Other instances include matters where computers have allegedly been employed to perpetrate a crime, fraud or tort, such as theft of trade secrets, workplace harassment, concealment of assets, hacking, theft of service, electronic vandalism, identity fraud, copyright infringement, etc.

Forensic Imaging Should Be Routine

Since it's not always possible to ascertain the need for computer forensic analysis at the onset of a dispute and with computer data being so volatile and fluid, how can a litigant preserve the status quo and protect potentially discoverable data? The best answer seems to be to act decisively to enforce the obligation to preserve while deferring disputes concerning the obligation to produce. At least with respect to the computer systems used by key players, if an opponent is unwilling to immediately remove them from service and secure them against tampering, loss or damage, then it is imperative that the hard drives for each computer be duplicated in a *forensically-sound* fashion and secured. They may never be used but, if needed, there is no better mechanism to demonstrate diligence in the preservation of discoverable data. The same prudence applies to other media which may later be claimed to have contained relevant and discoverable data, including personal digital assistants, e-mail servers and online repositories. *Caveat:* Routine file back up to tape, floppy disks, recordable CDs, thumb drives or other media using virtually any off-the-shelf back up application will *not* produce a forensically sound clone of the data, rendering some or all latent data unrecoverable in the future, ripe for a charge of spoliation.

Answers to Frequently Asked Questions about Forensic Imaging

What is a "forensically-sound" duplicate of a drive?

A "forensically-sound" duplicate of a drive is, first and foremost, one created by a method which does not alter data on the drive being duplicated. Second, a forensically-sound duplicate must contain a copy of every bit, byte and sector of the source drive, including unallocated "empty" space and slack space, precisely as such data appears on the source drive relative to the other data on the drive. Finally, a forensically-sound duplicate will not

contain any data (except known filler characters) other than which was copied from the source drive. All of this must be achieved in an authenticable way.

What's the difference between a "clone" and an "image" of a drive?
These terms are often used interchangeably, along with others like "bit stream copy," "mirror" and "ghost." So long as the duplicate is created in a forensically-sound way and can be reliably verified to be so, the name attached to the duplicate doesn't make much difference. However, the term "drive image" is most closely associated with a method of forensic duplication whereby all of the data structures on the source drive are stored in a file or series of files which, though structurally different from the source drive, can be reconstituted ("restored") in such a way as to be a forensically-sound duplicate of the source drive. A drive image is typically used with compression algorithms to store of the source drive data in a more compact fashion. Though a drive image is capable of being restored to create a clone drive, modern drive analysis software is designed to "virtually restore" the drive, reading directly from the image file and "seeing" the forensically-sound duplicate drive without the necessity for restoration.

How do you make a "forensically-sound" duplicate of a drive?
Although many forensic examiners use similar techniques and equipment, there is no one "recognized" or "approved" way to create a forensically-sound duplicate of a drive. There are a number of hardware and software tools well-suited to the task, each with their strengths and weaknesses, but all are capable of creating a forensically-sound duplicate of a typical PC hard drive when used correctly. Keep in mind that there are many different types of digital media out there, and a tool well-suited to one may be incapable of duplicating another. You simply have to know what you are doing and select he correct tools for the job

Duplication tools fall into two camps: those which create a drive image (a file which can be restored to match the source) and those which create a clone drive (a target drive or other media that duplicates the source data without the need for data restoration). My favored approach was once to clone drives but that has all but entirely given way to drive imaging. Again, done right, either approach works. Just get everything (including unallocated clusters and file slack) and be sure you can authenticate the duplicate.

To create forensically sound copies of hard drives, I've variously used a host of approaches, ranging from generic software capable of producing a bit stream duplicate to custom-built applications exclusively for forensic drive duplication to handheld devices that automate nearly the entire process. One alternative is a hardware cloning devices like those from Intelligent Computer Systems (www.ics-iq.com) or Logicube, Inc. (www.logicube.com). For high speed onsite acquisition, I like the ICS Image Masster Solo handheld drive duplication tools ($1,995.00) that allow me to simply hook up a source and target drive, push a few buttons and go. I've tested its accuracy using hash signature tools and, in every instance, the duplicate created by the Solo was forensically sound. I use the Solo in conjunction with a hardware-based write-blocking device called Drive Lock ($195.00), also from Intelligent Computer Systems, which intercepts any efforts by the Solo to write to the source drive. Since a tired or distracted user can accidentally swap the source and target drives,

irretrievably destroying the evidence, a hardware-based write blocker is an ideal way to be absolutely certain that the source drive will not be altered during the duplication process.

Other specialized duplication methods entail using forensic applications like Forensic Tool Kit Imager (nominally $89.00 but freely downloadable from Access Data; www.accessdata.com), EnCase Forensic Edition ($3,600.00 from Guidance Software, Inc.; www.guidancesoftware,com) or X-Ways Forensics ($584.00 from X-Ways Software Technology AG; www.x-ways.com) to create a drive image. These applications are designed expressly to support computer forensic examiners and are all excellent products. For a less-costly approach, consider Symantec's Norton Ghost ($69.95 from Symantec, Inc.; www.symantec.com) or the free Linux dd utility (included with any version of Linux). Ghost has been maligned as a forensic tool because, when used with its default commands and settings, it violates the cardinal rule of computer forensics—it changes data on the source drive. However, if Ghost is used with care—and the correct command line switches and settings are selected—it's capable of creating either a forensically-sound image or clone disk. If you're adept with the free Linux operating system, using Linux' dd (for disk dump) utility is surely the most cost effective solution. Here again, in untrained hands, dd is an unforgiving application and can destroy evidence; but, used with care by one who knows what they are doing, it's a gem.

There are many products on the market that claim to duplicate "everything" on a drive, but beware, as most are merely back up utilities and don't preserve the unallocated space. Unless the product carries over ever bit and sector of the source drive, without modification or corruption, it's wholly unsuited for computer forensics. Before settling on any duplication product, peruse the literature, solicit recommendations from computer forensic specialists and review test results at the National Institute of Standards and Technology's (NIST) Computer Forensic Tool Testing program (http://www.cftt.nist.gov/index.html).

How can you prove the duplicate drive is forensically sound?
Considering the size of modern hard drives, one way you *can't* prove the validity of your duplicate is by manually comparing the data. It's just impossible. So, the process of verification has got to be automated and foolproof. To appreciate the solution, take a moment to ponder the problem: how can you examine perhaps forty, sixty, eighty *billion* entries on a duplicate drive and be certain that every one of them has precisely the same value and is in the exact same relative location as on the source drive? Not just be *certain*, but be more reliably certain than fingerprints and more than DNA evidence. This is where we say "thanks" to all the mathematical geniuses who gave up normal human interaction to dedicate their lives to algorithms, arrays and one-way computations. These are the brainiacs who thought up "hash functions" and "message digests."

A hash function accepts a value of any size as its input, performs a complex calculation on that input and returns a value of fixed length as its output. The output value functions as a unique representation of the input. Put in a complex "message" and out pops a long string of letters and number bearing no discernable relationship to the message but which can only be generated by the one input. Accordingly, the output is called a "message digest." The really

amazing part of this is that the computation only works in one direction. It's considered "computationally infeasible" to decode the input from the output, which is a fancy way to say "Fuhgeddaboudit!" Since the input message can be anything, someone had the very bright idea to use the entire contents of a hard drive or thumb drive as the input and—voila!—the output becomes a fingerprint of that drive's contents and layout. Change so much as one single bit somewhere on the drive and the message digest changes dramatically. Since the fingerprint is unique to the inputted message (here, the data on the drive) only a forensically-sound duplicate of the drive could generate the same message digest.

Two widely-used hash functions are called MD5 and SHA-1. MD-5 generates a 32 character (128-bit) string that might look something like this: *9E2930D48131COFC9EE646AE2197A69C*. No matter how long or short the input, the MD5 output always is thirty-two characters in length. The chance of two different inputs producing the same MD5 message is greater than 1 in 340 undecillion. That's a staggering I in *340,000,000,000,000,000,000,000,000,000,000,000,000* chance! That beat's the pants off of DNA and fingerprints, and SHA-1 is even *more* reliable.

> In 2004, four Chinese researchers, Xiaoyun Wang, Dengguo Feng, Xuejia Lai and Hongbo Yu, succeeded in using a supercomputer to fabricate slightly different files with identical MD-5 hash values. Though still an excellent tool for validation, experts expect a gradual move away from MD-5 to even more secure hash algorithms.

Steps to Preserve the Evidence

A thorough exploration of the legal issues and precedents concerning the duty to preserve and produce electronically stored information is beyond the scope of this paper, but the near-term goal must be to preserve the status quo lest, like the lawyers litigating *Jarndyce v. Jarndyce* in Charles Dickens' "Bleak House," the lawyers keep squabbling until there is nothing left to fight over. Faced with a potential for forensic analysis, forensically sound duplication of potentially relevant media is key to preserving evidence.

As soon as it appears that computer data—and above all, latent data—may lead to the discovery of admissible evidence (or may meet whatever standard your jurisdiction applies to define what must be preserved), several things should be done:

1. The opposition should be expressly advised that the computer data is regarded as evidence and that immediate steps must be taken to preserve all such evidence until the court has an opportunity to address its discoverability. Because few people have a full appreciation of how much latent data exists on their machines or the adverse impact ongoing use can have on such data, you will need to be quite specific in your description of the actions to be taken or avoided, as well as in your identification of the target media. In some instances, you may be justifiably concerned that such a communiqué will serve as a road map to the destruction of evidence, but if you hope to have any chance of proving spoliation, you will need to be certain that ignorance won't serve as a defense. For further guidance in drafting a preservation notice, see the article entitled, "**The Perfect Preservation Letter"** at www.craigball.com.

2. Begin the process of educating the court about electronic evidence by moving for a protective order requiring that the party in possession of the computer refrain from any action that may impair the ability to recover latent or dynamic data. The goal initially is not to fight all the discovery battles, but only to preserve the status quo so that evidence doesn't disappear.

3. Secure two forensically sound duplicates of the evidence media. Once the accuracy has been established by hashing, you will want to leave one copy completely untouched and use the other for analysis to guard against any accusation that data was altered or corrupted during analysis. Hard drives are cheap. Sanctions are expensive. Preserve a chain of custody with respect to the copies or you will impair their usefulness. Be certain that the person selected to make the copies is fully qualified by training or experience to do so. You may be choosing a courtroom witness, so demeanor and appearance should play a role in your selection.

4. Seek an agreement with opposing counsel to engage, or get a court order to appoint, a special master to act as an impartial custodian of the original media and/or bit stream copies. Ideally, the special master should be both an attorney and skilled in computer forensics. It may not be necessary for the special master to be a computer forensics expert—he or she can hire skilled personnel as needed and supervise their work—but the master must be sufficiently conversant in all of the principal issues discussed in this article so as to be able to guide the court and communicate with technical personnel. Using a lawyer as the special master streamlines the identification and resolution of privilege, privacy, trade secret, relevance and discoverability issues. Some courts vest in the special master a limited authority to resolve discovery disputes within the ambit of the master's delegated responsibility. No matter how such matters are handled, the master's duty is to serve as an *impartial* custodian or arbiter, affording both sides a full and fair opportunity to have their concerns aired and their rights protected.

What's It Going to Cost?

Computer forensic analysis is exacting work requiring specialized knowledge, specialized tools, patience, tenacity, restraint, insight and no small measure of investigative talent. Analysts tend to come from the ranks of law enforcement or the military; but neither a working knowledge of forensic procedures nor an intimate acquaintance with computers alone suffice to qualify one as a computer forensic specialist. A competent forensic analyst needs both skill sets. That is, of course, a prelude to saying, "it's expensive."

Plan on paying from $150.00 to $500.00 per hour for forensic analysis and, while a quick-and-dirty, well-focused drive analysis might be completed in a day or two, a complex analysis can take much longer.

One area in which costs can never be cut is in the use of slipshod evidentiary procedures. No matter how convinced you might be that the information uncovered will never be offered in court, a competent forensic examiner won't do the job in a way that will taint the evidence. A

competent examiner never boots from the original drive. A competent examiner never "just takes a quick peek" at the data. A competent examiner never uses the original media in the analysis. *Never!*

Using a computer special master with a law degree, litigation experience and computer forensic ability is going to cost $350.00 to $550.00 or more per hour depending upon training, experience and stature, but the additional cost should be offset by a quicker resolution of discovery disputes and a diminished reliance upon the court acting *in camera.* The use of an impartial master with computer skills can also free the parties from having to hire their own computer forensic experts. For more on this, read, "**Finding the Right Computer Forensics Expert,"** at www.craigball.com.

Who Pays?

With the advent of electronic discovery, the longstanding presumption that a producing party bears the cost to identify, collect and bring forward material sought in discovery is increasingly being challenged by litigants and re-examined by courts. Meeting an e-discovery demand in the 21st century can be substantially more costly than its 20th century paper-centric counterpart. The higher cost of electronic discovery is a function of the greater volume, depth and complexity of electronic recordkeeping and a problem exacerbated by fundamental flaws in the way computers and users create and store digital information. The good news is that it's not always going to be more expensive and, when we finally get our digital acts together, e-discovery will be the *only* cost-effective solution. Until then, lawyers can look forward to years of quarreling over who pays. As a general proposition, the party seeking forensic analysis pays for that work unless the need for the effort arose because of malfeasance on the part of the other side (e.g., data destruction or alteration).

Is Digital Different?

Faced with a demand for cost shifting, the party seeking electronic discovery might wonder, "Why should the courts depart from the longstanding practice that the producing party pays? Should a requesting party be disadvantaged simply because an opponent has adopted an electronic mechanism for creating and storing information? *We're* not asking for more, *they're* just creating and keeping more of the stuff we seek! Didn't the producing party *choose* to computerize, voluntarily and for its own benefit? "

In fact, the stampede to computerization, with the attendant strain on discovery boundaries and budgets, was so broad and deep a sea change, why even call it a choice? We got where we are before anyone realized how far out on the limb we'd climbed. A device evolved from an electronic toy no one expected to succeed now sits on every desk and serves as the conduit for much of our communication, research, commerce, entertainment and misbehavior. Does the shift from paper to bits and bytes matter? Is digital different?

A business record born on paper (e.g., a handwritten form or a letter from a typewriter) is pretty much "what-you-see-is-what-you-get." There is no layer of information lurking within the fibers of the paper. You don't need special tools or techniques to glean the contents. A photocopy probably conveys about as much useful information as the original. Absent

forgery, the author and addressee are there in black-and-white. But, digital is different. Computer-generated documents all have *metadata* associated with them. That is, data about data, information outside the body of the document that conveys such things as when the document was created or modified, its author, electronic format, file size and more. Moreover, the creation of an electronic record often engenders the creation of a host of other records, some, like back up files or prior versions, the users knows about and some, like log, spool and swap files, the user may never imagine exist. Computers have also facilitated the recording of communications that not long ago would never have been reduced to writing. E-mail now stands in for conversations that would have been phone calls or water cooler chitchat a few years ago. The end result is that discoverable information exists in new planes, not only a broader swath of data, but a deeper level as well.

An exponential increase in discoverable volume is not the only challenge, nor is it the most difficult to resolve. A greater hurdle stems from the manner in which computers retain and dispose of information. Can you imagine a business that managed all its records and transactions—personal, professional, intimate, recreational, confidential and privileged--by dumping one and all into a big bin? How about a lawyer dumping every scrap of paper in her life--wills, bills, stills, frills and thrills--in a giant folder labeled "Stuff?" It's hard to image that level of incompetence, but we'd certainly expect that such malfeasance--commingling client materials with personal and third party stuff--would hobble claims of privilege or confidentiality. Yet, that's what a computer routinely does in its management of swap files, e-mail folders and the web surfing cache, to name just a few problem areas.

If that's not bad enough, the computer is a trash can without a bottom! You try to tidy up by deleting files and the computer just hides them (or pieces of them) from you, squirreling data away like acorns, willy-nilly, across a vast expanse of hard drive! Is it any wonder that trying to makes sense of this mess is expensive?

Lawyers frequently approach e-discovery as they've always done with paper records. But we've had thousands, of years to master the management of paper records, and the innate physicality of a writing means it's easier to track, isolate and, ultimately, destroy. Digital is different, and, while the rules of procedure and evidence may prove sufficiently flexible to adapt to a virtual world, some in the bench and bar loathe straying far from their familiar, paper-based systems. Inflexibility boosts the cost of electronic discovery, through, *inter alia*, the use of tortured definitions of "document" in discovery requests, impossibly overbroad production demands, compulsory "blow back" of native digital data to paper printouts (with the attendant loss of the metadata layer). More costly still is the practice of reducing data to a page-like format to facilitate privilege review. When even a modestly-sized hard drive can easily generate a million "pages" of documents and a server, tens- or hundreds of millions of pages, there are simply not enough eyeballs that can be placed in front of enough desks to complete the job in the customary fashion. Because digital is different, we must change as well.

Shifting Costs: The Rowe and Zubulake Decisions

Though this discussion has steered wide of the burgeoning case law governing electronic discovery, one can't talk about planning for the cost of computer forensics in e-discovery without at least touching on the two most important decisions on the topic: *Rowe Entertainment, Inc. v. The William Morris Agency, Inc.*, 205 F.R.D. 421 (S.D.N.Y. 2002) and *Zubulake v. UBS Warburg LLC*, F.R.D. 309 (S.D.N.Y. 2003).

The import of these decisions is that they articulate factors to be weighed by a court in determining whether the cost of responding to a discovery request should be shifted to the party seeking discovery. The *Rowe* opinion put forward eight factors, but proved to favor disproportionately large entities resisting discovery. Accordingly, the Court in *Zubulake*--a discrimination case where the plaintiff sought e-mail stored on back up tapes—re-visited the *Rowe* factors and derived a three-part approach to determining whether cost-shifting is appropriate.

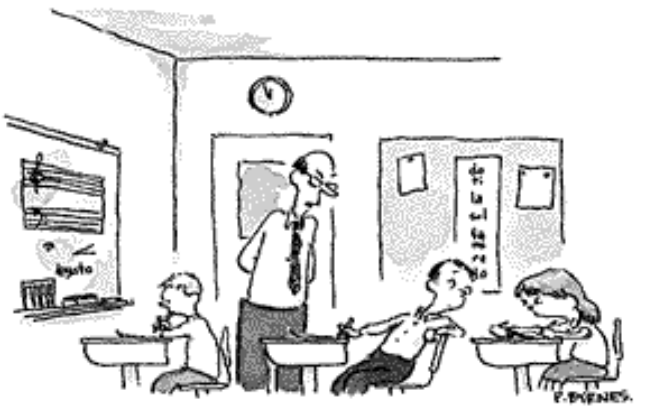
Significantly, the *Zubulake* court makes clear that if the materials sought are "accessible" (e.g., active online data or readily available near-online data like optical disks), the responding party bears the cost of production absent undue burden or expense warranting protection. However, if the materials sought are inaccessible--such as e-mail on legacy back up tapes and most information developed through forensic examination--the Court may consider cost shifting and undertake a factual inquiry to identify what type of information is likely to reside on the "inaccessible" media. This inquiry may entail some sampling of the inaccessible media to gauge its relevance and the level of cost and effort in meeting the discovery request. Finally, as the third leg of the analysis, the Court set out seven factors to be used in balancing interests and burdens. In the order of importance which the Court ascribed to them, the seven considerations are:

(1) Is the request specifically tailored to discover relevant information?
(2) Is the information available from other sources?
(3) How does cost of production compare to the amount in controversy?
(4) What are the relative positions of the parties in terms of resources?
(5) Who is best able to control costs and has an incentive to do so?
(6) Are the issues in discovery key to the issues at stake in the litigation?
(7) What are the relative benefits to the parties of obtaining the data?

The Court's recognition of sampling as an appropriate means to gauge the likelihood that discovery will prove fruitful enough to justify the attendant burden is noteworthy. Though the Zubulake court set the sample size, it left the selection of the items sampled to the party seeking discovery. While this introduces an element of happenstance, unless the tools of discovery better tame the volume, sampling is probably as sound a splitting of the baby as any other. Another notable aspect of the decision is the Court's refusal to shift the cost of privilege review to the requesting party, reasoning that the producing party is better situated to control this cost and that, once inaccessible data is restored for review, it's really no

different than any other discovery materials and, accordingly, review costs would ordinarily be borne by the producing party.

The Court did not address cost shifting when forensic intervention is sought in response to a producing party's obstructive or destructive actions, such as failing to preserve electronic evidence or affirmative efforts to eliminate same. In those circumstances, Courts are likely to visit all the costs of discovery upon the producing party but intervene to protect the rights of third-parties and preserve privilege.

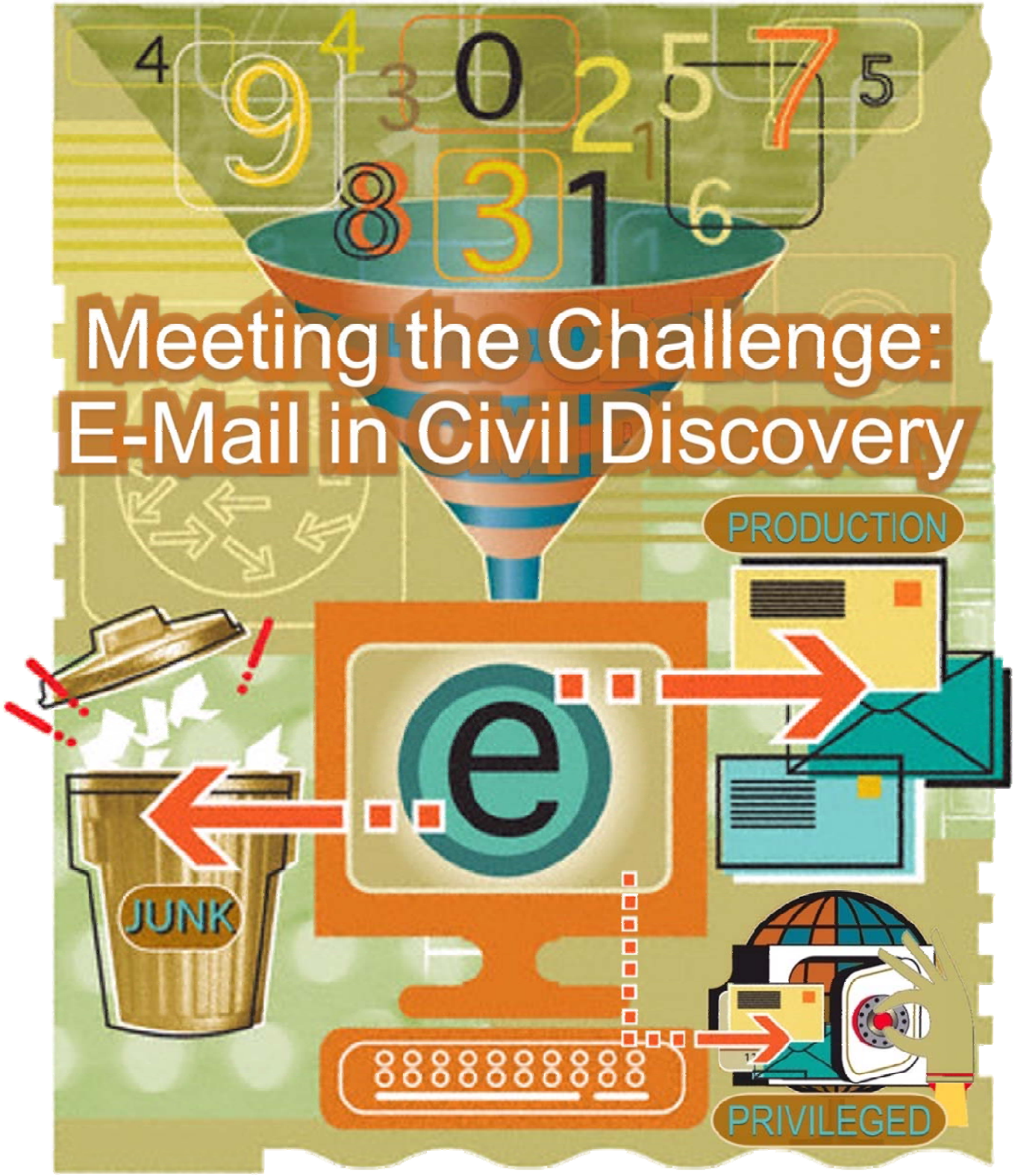


"Mister Jackson! You know how I feel about sampling."

The Rough Road Ahead

The next decade will see the introduction of a wondrous array of new and sophisticated technology tools and toys. Hard drives will continue to grow in capacity and drop in price per gigabyte-- expect to start seeing a terabyte of storage in PCs before the end of 2008. Wireless connectivity will be ubiquitous and online storage will grow in importance. Personal digital assistants will continue to converge with cellular phones, MP3 players and global positioning devices, exemplified by the Blackberry, Palm Treo and Apple iPhone, Matchbook-sized hard drives and tiny high capacity portable media like thumb drives will find their way into a host of new gadgets, many with unique, proprietary operating systems. We will continue to see increased reliance on and integration of computers in our lives. These machines will look less and less like our current clunky desktops, and they will be nimbler and more specialized than the personal computer we see today. Greater portions of our daily lives and labors will be recorded digitally and stored online in richer media formats like sound and video. Paper will not disappear, but little of what we deal with on paper today will remain in paper form. Encryption will be easier to use and will be built into more applications that create and store information.

It sounds pretty exciting and positive—and it is--but the dark side for litigators is that discovery of electronic evidence is not only going to become a larger part of our practice, it's going to get harder and cost more. We will be seeking discovery of data stored in cell phones, automobile dashboards and personal stereos. Cherished notions of personal privacy will continue to collide with our growing ability to track, record, analyze, communicate and compile personal information. It will be challenging, to say the least, and it requires lawyers to cultivate an understanding of technology as never before; but, if you've read this far and "get it" (or most of it), you're someone who can turn the coming challenges into opportunities.



Meeting the Challenge: E-Mail in Civil Discovery
Craig Ball

Table of Contents

Understanding E-Mail in Civil Discovery

Introduction

Get the e-mail! It's the war cry in discovery today. Some label the press for production of electronic mail a feeding frenzy, but it's really just an inevitable recognition of e-mail's importance and ubiquity. Lawyers go after e-mail because it accounts for the majority of business communication, and because e-mail users tend to let their guard down and share things online that they'd never dare put in a memo. But if you're the lawyer who'll be on the receiving end of a request for production, you not only have to be concerned about the contents of the messages, you face the earlier, bigger challenge of finding your client's e-mail, preserving it from spoliation and producing responsive items without betraying privileged or confidential communications. Meeting that challenge effectively requires that understanding e-mail technology well enough to formulate a sound, defensible strategy.

This paper seeks to equip the corporate counsel or trial lawyer with much of what they need to know to pursue or defend against the discovery of e-mail in civil litigation. Be warned that the paper is replete with technical information which I've tried to convey in manner that anyone reasonably comfortable with personal computers can grasp. If you don't enjoy technical topics, I urge you to plow through anyway because it's so important for a litigator to have a working knowledge of computer technology. Your "reward" will be the forty tips at the conclusion. Hopefully the tips, and the other information that follows, will help you be better prepared to meet the e-mail challenge.

Not Enough Eyeballs

Futurist Arthur C. Clarke said, "Any sufficiently advanced technology is indistinguishable from magic." E-mail, like electricity, refrigeration and broadcasting, is one of those magical technologies most of use every day without really understanding how it works. But is there a judge who will accept, "it's just magical," as an explanation of your client's e-mail system or as justification for a failure to preserve or produce discoverable e-mail?

A lawyer managing electronic discovery is obliged to do more than just tell their clients to "produce the e-mail." You've got to make an effort to understand their systems and procedures and ask the right questions, as well as know when you aren't getting the right answers. That's asking a lot, but 95% of all business documents are born digitally and few are ever printed. Almost *seventy billion* e-mails traverse the Internet *daily,* far more than telephone and postal traffic combined, and the average business person sends and receives between 50 and 150 e-mails *every business day.* E-mail contributes *500 times greater volume* to the Internet than web page content. In discovery, it's increasingly infeasible to put enough pairs of trained eyes in front of enough computers to thoroughly review every e-mail. Much as we might like, we lawyers can't put our heads under our pillows and hope that it all goes away. The volume keeps increasing, and there's no end in sight.

Test Your E.Q.

While I'm delivering bad news, let me share *worse* news: if you don't change the way your business clients do business by persuading them to initiate and enforce web- and e-mail usage restrictions with an iron fist, complete with Big Brother-style monitoring, you *will* fail to locate and produce a sizable part of your clients' electronic communications--and you won't even *know* you missed them until you see examples attached to opposing counsel's motion for sanctions. Of course, e-mail enabled cell phones like the Blackberry, Treo and other PDAs present pitfalls, but I'm also alluding to the digital channels that fall *outside* your client's e-mail server and back up tape system, like Instant Messaging, browser based e-mail and voice messaging.

Suppose opposing counsel serves a preservation letter or even a restraining order requiring your client to preserve electronic messaging. You confidently assure opposing counsel and the court that your client's crack team of information technologists will faithfully back up and preserve the data on the e-mail servers. You're more tech-savvy than most, so you even think to suspend the recycling of back up tapes. But are you really capturing all of the discoverable communications? How much of the 'Net is falling outside your net?

Can you answer these questions about your client's systems?

- Do *all* discoverable electronic communications come in and leave via the company's e-mail server?
- Does your client's archival system capture e-mail stored on individual user's hard drives, including company-owned laptops?
- Do your clients' employees use personal e-mail addresses or browser-based e-mail services (like Gmail or Yahoo Mail) for business communications?
- Do your clients' employees use Instant Messaging on company computers or over company-owned networks?
- How do your clients' voice messaging systems store messages, and how long are they retained?

Troubled that you can't answer some of these questions? You should be, but know you're not alone. If your client runs a large network, capturing all the messaging traffic is a

challenge akin to catching a spilled bucket of water in your bare hands. It's nearly impossible, and *you are going to miss something*.

Staying Out of Trouble

Fortunately, the rules of discovery don't require you to do the impossible. All they require is diligence, reasonableness and good faith. To that end, you must be able to establish that you and your client acted swiftly, followed a sound plan, and took such action as reasonable minds would judge adequate to the task. It's also important to keep the lines of communication open with the opposing party and the court, seeking agreement with the former or the protection of the latter where fruitful. I'm fond of quoting Oliver Wendall Holmes' homily, "Even a dog knows the difference between being stumbled over and being kicked." Likewise, it's hard to get much traction for a sanctions motion when it is clear to all concerned that the failure to produce electronic evidence was not part of an effort to conceal something or grew out of laziness, stupidity or arrogance.

…And You Could Make Spitballs with It, Too

Paper discovery enjoyed a self-limiting aspect in that businesses tended to allocate paper records into files, folders and cabinets according to persons, topics, transactions or periods of time, and did so throughout the business process. The space occupied by paper and the high cost to create, manage and store paper records served as a constant impetus to cull and discard them, or even to avoid creating them in the first place. By contrast, the ephemeral character of electronic communications, the ease of and perceived lack of cost to create, duplicate and distribute them and the very low direct cost of data storage has facilitated a staggering and unprecedented growth in the creation and retention of electronic evidence. At fifty e-mails per day, a company employing 100,000 people could find itself storing well over *1.5 billion* e-mails annually.

Did You Say *Billion*?

But volume is only part of the challenge. Unlike paper records, e-mail tends to be stored in massive data blobs. The single file containing my Outlook e-mail is almost three gigabytes in size and contains tens of thousands of messages, many with multiple attachments, covering virtually every aspect of my life, and many other people's lives, too. In thousands of those e-mails, the subject line bears only a passing connection to the contents as "Reply to" threads strayed further and further from the original topic. E-mails meander through disparate topics or, by absent-minded clicks of the "Forward" button, lodge in my inbox dragging with them, like toilet paper on a wet shoe, the unsolicited detritus of other people's business. To respond to a discovery request for e-mail on a particular topic, I'd either need to skim/read countless messages or I'd have to have a very high degree of confidence that a keyword search would flush out all responsive material. If the request for production implicated material I no longer kept on my current computer, I'd be forced to root around through a motley array of archival folders, old systems, obsolete disks, outgrown hard drives, ancient back up tapes (for which I have no tape reader) and unlabeled CDs, uncertain whether I've lost the information or just overlooked it somewhere along the way.

Net Full of Holes

So what's a company to do when served with a request for "all e-mail" on a particular matter in litigation? Surely, I mused, someone must have found a better solution than repeating, over and over again, the tedious and time-consuming process of accessing individual e-mail servers at far-flung locations along with the local drives of all key players' computers? For this article, I contacted colleagues in both large and small electronic discovery consulting groups, inquiring about "the better way" for enterprises, and was struck by the revelation that, if there was a better mousetrap, they hadn't discovered it either. Uniformly, we recognized such enterprise-wide efforts were gargantuan undertakings fraught with uncertainty, and concluded that counsel must somehow seek to narrow the scope of the inquiry—either by data sampling or through limiting discovery according to offices, regions, time span, business sectors or key players. Trying to capture *everything,* enterprise-wide, is trawling with a net full of holes.

E-Mail Systems and Files

Michelle Lange of the national e-discovery firm Kroll OnTrack relates that Microsoft Exchange Server and Outlook e-mail account for nearly 75% of the e-mail Kroll encounters in its engagements, with Lotus Notes a distant second at 13%. Accordingly, the following discussion principally addresses the Microsoft e-mail applications, but be aware that each system employs its own twist on file structures and names. For example, AOL has long used a proprietary mail format incompatible with other common standards.

A Snippet about Protocols

Computer network specialists are always talking about this "protocol" and that "protocol." Don't let the geek-speak get in the way. An *application protocol* is a bit of computer code that facilitates communication between applications, i.e., your e-mail client, and a network like the Internet. When you send a snail mail letter, the U.S. Postal Service's "protocol" dictates that you place the contents of your message in an envelope of certain dimensions, seal it, add a defined complement of address information and affix postage to the upper right hand corner of the envelope adjacent to the addressee information. Only then can you transmit the letter through the Postal Service's network of post offices, delivery vehicles and postal carriers. Omit the address, the envelope or the postage--or just fail to drop it in the mail--and Grandma gets no Hallmark this year! Likewise, computer networks rely upon protocols to facilitate the transmission of information. You invoke a protocol—*Hyper Text Transfer Protocol*—every time you type *http://* at the start of a web page address.

Incoming Mail: POP, IMAP, MAPI and HTTP E-Mail

Although Microsoft Exchange Server rules the roost in enterprise e-mail, it's by no means the most common e-mail system for the individual and small business user. When you access your personal e-mail from your own Internet Service Provider (ISP), chances are your e-mail comes to you from your ISP's e-mail server in one of three ways, POP, IMAP or HTTP, the last commonly called web- or browser-based e-mail. Understanding how these three protocols work—and differ—helps in identifying where e-mail can or cannot be found.

POP (for Post Office Protocol) is the oldest and most common of the three approaches and the one most familiar to users of the Outlook Express, Netscape and Eudora e-mail clients.

Using POP, you connect to a mail server, download copies of all messages and, unless you have configured your e-mail client to leave copies on the server, the e-mail is deleted on the server and now resides on the hard drive of the computer you used to pick up mail. Leaving copies of your e-mail on the server seems like a great idea, since you have a back up if disaster strikes and can access your e-mail, again and again, from different computers. However, few ISPs afford unlimited storage space on their servers for users' e-mail, so mailboxes quickly become "clogged" with old e-mails and the servers start bouncing new messages. As a result, POP e-mail typically resides only on the local hard drive of the computer used to read the mail and on the back up system for the servers which transmitted, transported and delivered the messages. In short, POP is locally-stored e-mail that supports some server storage.

IMAP (Internet Mail Access Protocol) functions in much the same fashion as most Microsoft Exchange Server installations in that, when you check your e-mail, your e-mail client downloads just the headers of e-mail it finds on the server and only retrieves the body of a message when you open it for reading. Else, the entire message stays in your account on the server. Unlike POP, where e-mail is searched and organized into folders locally, IMAP e-mail is organized and searched on the server. Consequently, the server (and its back up tapes) retains not only the messages but also the way the user structured those messages for archival. Since IMAP e-mail "lives" on the server, how does a user read and answer it without staying connected all the time? The answer is that IMAP e-mail clients afford users the ability to synchronize the server files with a local copy of the e-mail and folders. When an IMAP user reconnects to the server, local e-mail stores are updated (synchronized) and messages drafted offline are transmitted. So, to summarize, IMAP is server-stored e-mail, with support for synchronized local storage.

MAPI (Messaging Application Programming Interface) is the e-mail protocol at the heart of Microsoft's Exchange Server application. Like IMAP, MAPI e-mail is typically stored on the server, not the client machine. Likewise, the local machine may be configured to synchronize with the server mail stores and keep a copy of mail on the local hard drive, but this is user- and client application-dependent. If the user hasn't taken steps to keep a local copy of e-mail, e-mail is not likely to be found on the local hard drive, except to the extent fragments may turn up through computer forensic examination.

HTTP (Hyper Text Transfer Protocol) mail, or web-based/browser-based e-mail, dispenses with the local e-mail client and handles all activities on the server, with users managing their e-mail using their Internet browser to view an interactive web page. Although some browser-based e-mail services support local synchronization with an e-mail client, typically users do not have any local record of their browser-based e-mail transactions except for messages they've affirmatively saved to disk or portions of e-mail web pages which happen to reside in the browser's cache (e.g., Internet Explorer's Temporary Internet Files folder). Hotmail and Yahoo Mail are two popular examples of browser-based e-mail services, although many ISPs (including all the national providers) offer browser-based e-mail access in addition to POP and IMAP connections.

The protocol used to carry e-mail is not especially important in electronic discovery except to the extent that it signals the most likely place where archived e-mail can be found. Companies choose server-based e-mail systems (e.g., IMAP and MAPI) for two principal reasons. First, such systems make it easier to access e-mail from different locations and machines. Second, it's easier to back up e-mail from a central location. Because IMAP and MAPI systems store all e-mail on the server, the back up system used to protect server data can yield a mother lode of server e-mail. Depending upon the back up procedures used, access to archived e-mail can prove a costly and time-consuming task or a relatively easy one. The enormous volume of e-mail residing on back up tapes and the potentially high cost to locate and restore that e-mail makes discovery of archived e-mail from back up tapes a big bone of contention between litigants. In fact, most reported cases addressing cost-allocation in e-discovery seem to have been spawned by disputes over e-mail on server back up tapes.

Outgoing Mail: SMTP and MTA

Just as the system that brings water into your home works in conjunction with a completely different system that carries wastewater away, the protocol that delivers e-mail to you is completely different from the one that transmits your e-mail. Everything discussed in the preceding paragraph concerned the protocols used to *retrieve* e-mail from a mail server. Yet, another system altogether, called SMTP for Simple Mail Transfer Protocol, takes care of outgoing e-mail. SMTP is indeed a very simple protocol and doesn't even require authentication, in much the same way as anyone can anonymously drop a letter into a mailbox. A server that uses SMTP to route e-mail over a network to its destination is called a Message Transfer Agent (MTA). Examples of MTAs you might hear mentioned by IT professionals include Sendmail, Exim, Qmail and Postfix. Microsoft Exchange Server is an MTA, too. In simplest terms, an MTA is the system that carries e-mail between e-mail servers and sees to it that the message gets to its destination. Each MTA reads the code of a message and determines if it is addressed to a user in its domain and, if not, it passes the message on to the next MTA after adding a line of text to the message identifying the route to later recipients. If you've ever set up an e-mail client, you've probably had to type in the name of the servers handling your outgoing e-mail (perhaps *SMTP.yourISP.com*) and your incoming messages (perhaps *mail.yourISP.com* or *POP.yourISP.com*).

Anatomy of an E-Mail Header

Now that we've waded through the alphabet soup of protocols managing the movement of an e-mail message, let's take a look inside the message itself. Considering the complex systems on which it lives, an e-mail is astonishingly simple in structure. The Internet protocols governing e-mail transmission require electronic messages to adhere to rigid formatting, making individual e-mails fairly easy to dissect and understand. The complexities and headaches associated with e-mail don't really attach until the e-mails are stored and assembled into databases and post office files.

An e-mail is just a plain text file. Though e-mail can be "tricked" into carrying non-text binary data like application files (i.e., a Word document) or image attachments (e.g., GIF or .JPG files), this piggybacking requires binary data be encoded into text for transmission. Consequently, even when transmitting files created in the densest computer code, *everything in an e-mail is plain text*.

Figure 1 shows the source code of an e-mail which I sent using a browser-based Hotmail account. The e-mail was sent from forensicguru@hotmail.com and addressed to craig@ball.net, with a cc: to ball@sbot.org. A small photograph in JPG format was attached to the message.

Before we dissect the e-mail message in Figure 1, note that any e-mail can be divided into two parts, the header and body of the message. By design, the header details the journey taken by the e-mail from origin to destination; but be cautioned that it's a fairly simple matter for a hacker to spoof (falsify) the identification of all but the final delivery server. Accordingly, where the origin or origination date of an e-mail is suspect, the actual route of the message may need to be validated at each server along its path.

In an e-mail header, each line which begins with the word "Received:" represents the transfer of the message between or within systems. The transfer sequence is reversed chronologically; such that those closest to the top of the header were inserted after those that follow, and the topmost line reflects delivery to the recipient's e-mail server. As the message passes through intervening hosts, each adds its own identifying information along with the date and time of transit.

Tracing an E-Mail's Incredible Journey

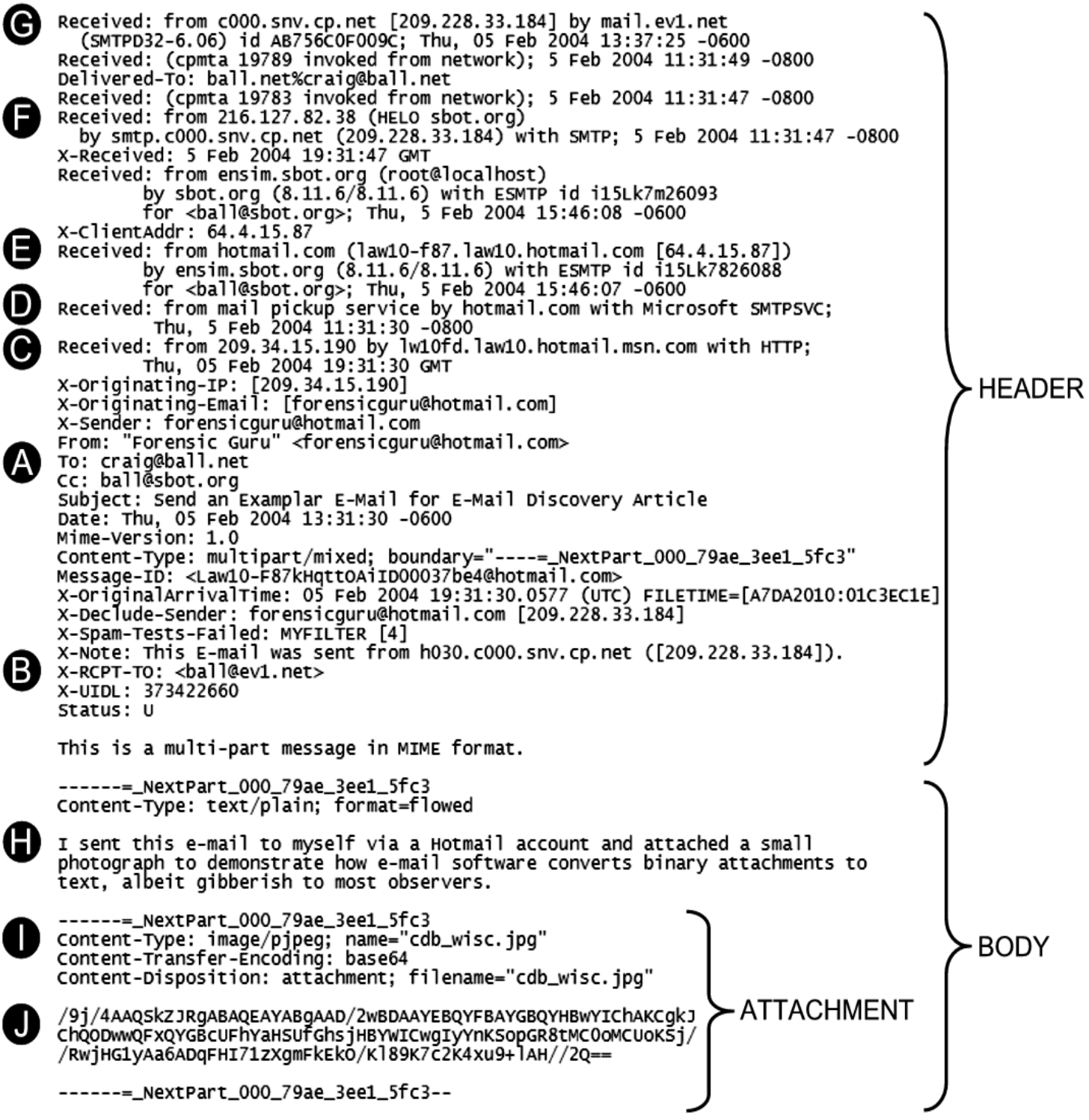
In this header, taken from the cc: copy of the message, section **(A)** indicates the parts of the message designating the sender, addressee, cc: recipient, date, time and subject line of the message. Though a message may be assigned various identification codes by the servers it transits in its journey (each enabling the administrator of the transiting e-mail server to track the message in the server logs), the message will contain one unique identifier assigned by the originating Message Transfer Agent. The unique identifier assigned to this message (in the line labeled "Message-ID:") is "Law10-F87kHqttOAiID00037be4@ hotmail.com." In the line labeled "Date," both the date and time of transmittal are indicated. The time indicated is 13:31:30, and the "-0600" which follows this time designation denotes the time *difference* between the sender's local time (the system time on the sender's computer) and Greenwich Mean Time (GMT), also called Universal Time or UTC. As the offset from GMT is minus six hours, we deduce that the message was sent from a machine set to Central Standard Time, giving some insight into the sender's location. Knowing the originating computer's time and time zone can occasionally prove useful in demonstrating fraud or fabrication.

At **(B)** we see that although this carbon copy was addressed to ball@sbot.org, the ultimate recipient of the message was ball@EV1.net. How this transpired can be deciphered from the header data.

The message was created and sent using Hotmail's web interface; consequently the first hop **(C)** indicates that the message was sent using HTTP from my home network router, identified by its IP address: 209.34.15.190. The message is received by the Hotmail server **(D),** which transfers it to a second Hotmail server using SMTP. The first Hotmail server timestamps the message in Greenwich Mean Time (GMT) but the second Hotmail server

timestamps in its local time, noting a minus eight hour offset from GMT. This suggests that the Hotmail server is located somewhere in the Pacific Time zone. The next hand off **(E)** is to

Figure 1.



the Ensim appliance on the SBOT.org server, where the message is designated for user ball@sbot.org. Note the erroneous timestamp affixed by the SBOT.org. Although the message has apparently come back into the Central Time zone, the receiving server's clock is some 135 minutes fast!

The message has reached its appointed destination at SBOT.org; however, its incredible journey is far from done. The header informs us that the SBOT.org server is set up to forward mail addressed to ball@sbot.org to another address, and so we follow the message as it heads to a server two time zones west, belonging to a company called Critical Path (cp.net). There, **(F)** the message is delivered to the address craig@ball.net. But it appears that mail addressed to craig@ball.net is also automatically forwarded to yet another address and server! The message skedaddled back to the Lone Star State, to a server operated by EV1.net, and **(G)** ultimately to the mailbox for ball@EV1.net **(B).**

Turning to the body of the message, notice how the content of the message **(H)** is set off from the header and the attachment **(I)** by a blank line and a boundary code generated by the e-mail client: **------=_NextPart_000_79ae_3ee1_5fc3**. Note, also, how the attachment, a photograph with the filename "cdb_wisc.jpg," has been encoded from non-printable binary code into a long string of plain text characters (J) able to traverse the network as an e-mail, yet easily converted back to binary data when the message reaches its destination. In order to fit the page, only three lines of the encoded data are shown. The encoded data actually occupied fifty lines of text.

Clearly, e-mail clients don't share onscreen all the information contained in a message's source but instead parse the contents into the elements we are most likely to want to see: To, From, Subject, body, and attachment. If you decide to try a little digital detective work on your own e-mail, you'll find that e-mail client software doesn't make it easy to see complete header information. In Microsoft Outlook Express, highlight the e-mail item you want to analyze and then select "File" from the Menu bar, then "Properties," then click the "Details" tab followed by the "Message Source" button. Think that sounds complicated? Microsoft's Outlook mail client makes it virtually impossible to see the complete message source; however, you can see message headers for individual e-mails by opening the e-mail then selecting "View" followed by "Options" until you see the "Internet headers" window on the Message Option menu.

Local E-Mail Storage Formats and Locations

Suppose you're faced with a discovery request for a client's e-mail, or you simply want to back up your own e-mail for safekeeping. Where are you going to look to find stored e-mail, and what form will that storage take? Because an e-mail is just a text file, individual e-mails could be stored as discrete text files. But that's not a very efficient or speedy way to manage a large number of messages, so you'll find that e-mail client software doesn't do that. Instead, e-mail clients employ proprietary database files housing e-mail messages, and each of the major e-mail clients uses its own unique format for its database. Some programs encrypt the message stores. Some applications merely display e-mail housed on a remote server and do not store messages locally (or only in fragmentary way). The only way to know with certainty if e-mail is stored on a local hard drive is to look for it. Merely checking the e-

mail client's settings is insufficient because settings can be changed. Someone not storing server e-mail today might have been storing it a month ago. Additionally, users may create new identities on their systems, install different client software, migrate from other hardware or take various actions resulting in a cache of e-mail residing on their systems without their knowledge. *If they don't know it's there, they can't tell you it's not*. On local hard drives, you've simply got to know what to look for and where to look…*and then you've got to look for it.*

For many, computer use is something of an unfolding adventure. One may have first dipped her toes in the online ocean using browser-based e-mail or an AOL account. Gaining computer-savvy, she may have signed up for broadband access or with a local ISP, downloading e-mail with Netscape Messenger or Microsoft Outlook Express. With growing sophistication, a job change or new technology at work, the user may have migrated to Microsoft Outlook or Lotus Notes as an e-mail client. Each of these steps can orphan a large cache of e-mail, possibly unbeknownst to the user but still fair game for discovery. Again, you've simply got to know what to look for and where to look.

One challenge you'll face when seeking stored e-mail is that every user's storage path can be, and usually is, different. This difference is not so much the result of a user's ability to specify the place to store e-mail—which few do, but which can make an investigator's job more difficult when it occurs—but more from the fact that operating systems are designed to support multiple users and so must assign unique identities and set aside separate storage areas for different users. Even if only one person has used a Windows computer, the operating system will be structured at the time of installation so as to make way for others. Thus, finding e-mail stores will hinge on your knowledge of the User Account or Identity assigned by the operating system. This may be as simple as the user's name or as obscure as {721A17DA-B7DD-4191-BA79-42CF68763786}. Customarily, it's both.

Caveat*: Before you or anyone on your behalf "poke around" on a computer system seeking a file or folder, recognize that absent the skilled use of specialized tools and techniques, such activity will result in changing data on the drive. Some of the changed data may be forensically significant (such as file access dates) and could constitute spoliation of evidence. If, under the circumstances of the case or matter, your legal or ethical obligation is to preserve the integrity of electronic evidence, then you and your client may be obliged to entrust the search only to a qualified computer forensic examiner.*

Finding Outlook Express E-Mail

Outlook Express has been bundled with every Windows operating system for nearly a decade, so you are sure to find at least the framework of an e-mail cache created by the program. However, since nearly everyone has Outlook Express but not everyone uses it (or sticks with it), finding Outlook Express mail stores doesn't tell you much about their contents.

Outlook Express places e-mail in files with the extension .dbx. The program creates a storage file for each e-mail storage folder that it displays, so expect to find at least Inbox.dbx, Outbox.dbx, Sent Items.dbx and Deleted Items.dbx. If the user has created other folders to hold e-mail, the contents of those folders will reside in a file with the structure

foldername.dbx. Typically on a Windows XP/NT/2K system—and I emphasize that each situation is unique—you will find Outlook Express .dbx files in the path from the root directory (C:\ for most users) as follows: **C:\Documents and Settings*useraccount*\Local Settings\Application Data\Identities\{*unique identifier string*}\Microsoft\Outlook Express**. Multiple identifier strings listed in the Identities subfolder may be an indication of multiple e-mail stores and/or multiple users of the computer. You will need to check each Identity's path. Another approach is to use the Windows Search function to find all files ending .dbx, but be very careful to enable all three of the following Advanced Search options before running a search: Search System Folders, Search Hidden Files and Folders, and Search Subfolders. If you don't, you won't find any—or at least not all—Outlook Express e-mail stores. Be certain to check the paths of the files turned up by your search as it can be revealing to know whether those files turned up under a particular user identity, in Recent Files or even in the Recycle Bin!

Finding Netscape E-Mail

Though infrequently seen today, Netscape and its Mozilla e-mail client ruled the Internet before the browser wars left it crippled and largely forgotten. If you come across a Netscape e-mail client installation, keep in mind that the location of its e-mail stores will vary depending upon the version of the program installed. If it is an older version of the program, such as Netscape 4.x and a default installation, you will find the e-mail stores by drilling down to **C:\Program Files\Netscape\Users*your profile name*\Mail**. Expect to find two files for each mailbox folder, one containing the message text with no extension (e.g., Inbox) and another which serves as an index file with a .snm extension (e.g., Inbox.snm).

In the last version of Netscape to include an e-mail client (Netscape 7.x), both the location and the file structures/names were changed. Drill down to **C:\Documents and Settings*Windows account name*\Application Data\Mozilla\Profiles\default*profile*.slt\Mail** and locate the folder for the e-mail account of interest, usually the name of the e-mail server from which messages are retrieved. If you don't see the Application Data folder, go to the Tools Menu, pull down to Folder Options, click on the View tab, and select "Show Hidden Files and Folders," then click "OK." You should find two files for each mailbox folder, one containing the message text with no extension (e.g., Sent) and another which serves as an index file with a .msf extension (e.g., Sent.msf). If you can't seem to find the e-mail stores, you can either launch a Windows search for files with the .snm and .msf extensions (e.g. *.msf) or, if you have access to the e-mail client program, you can check its configuration settings to identify the path and name of the folder in which e-mail is stored.

Finding Outlook E-Mail

Microsoft Outlook is by far the most widely used e-mail client in the business environment. Despite the confusing similarity of their names, Outlook is a much different and more complex application that Outlook Express. One of many important differences is that where Outlook Express stores messages in plain text, Outlook encrypts messages, albeit with a very weak form of encryption. But the most significant challenge Outlook poses in discovery is the fact that all of its local message data and folder structure, along with all other information managed by the program (except a user's Contact data), is stored within a single, often massive, database file with the file extension .pst. The Outlook .pst file format is proprietary

and its structure poorly documented, limiting your options when trying to view its contents to Outlook itself or one of a handful of .pst file reader programs available for purchase and download via the Internet.

To find the Outlook message store running Windows XP, NT or 2000, go to C:\Documents and Settings*windows user name*\Local Settings\Application Data\Microsoft\Outlook\Outlook.pst. The default filename of Outlook.pst may vary if a user has opted to select a different designation or maintains multiple e-mail stores; however, it's rare to see users depart from the default settings. Since the location of the .pst file can be changed by the user, it's a good idea to do a search of all files and folders to identify any files ending with the .pst extension.

Finding E-Mail on Exchange Servers

150 million people get their e-mail via a Microsoft product called Exchange Server. Though the preceding paragraphs dealt with finding e-mail stores on local hard drives, in disputes involving medium- to large-sized businesses, the e-mail server is likely to be the principal focus of electronic discovery efforts. The server is a productive venue in electronic discovery for many reasons, among them:

- Periodic back up procedures, which are a routine part of prudent server operation, tend to shield e-mail stores from those who, by error or guile, might delete or falsify data on local hard drives.
- The ability to recover deleted mail from archival server back ups may obviate the need for costly and sometimes fruitless forensic efforts to restore lost messages.
- Data stored on a server is often less prone to tampering by virtue of the additional physical and system security measures typically dedicated to centralized computer facilities as well as the inability of the uninitiated to manipulate data in the more-complex server environment.
- The centralized nature of an e-mail server affords access to many users' e-mail and may lessen the need for access to workstations at multiple business locations or to laptops and home computers.
- Unlike e-mail client applications, which store e-mail in varying formats and folders, e-mail stored on a server can usually be located with ease and adheres to a common file format.
- The server is the crossroads of corporate electronic communications and the most effective chokepoint to grab the biggest "slice" of relevant information in the shortest time, for the least cost.

Of course, the big advantage of focusing discovery efforts on the mail server (i.e., it can deliver up thousands or millions of messages) is also its biggest disadvantage (someone has to *extract and review* thousands or millions of messages). Absent a carefully-crafted and, ideally, agreed-upon plan for discovery of server e-mail, both requesting and responding parties run the risk of runaway costs, missed data and wasted time.

Server-based e-mail data is generally going to fall into two realms, being online "live" data, which is easily accessible, and offline "archival" data, which may be fairly inaccessible. Absent a change in procedure, "chunks" of data shift from the online to the offline realm on a

regular basis--daily, weekly or monthly—as selected information on the server is duplicated onto back up media and deleted from the server's hard drives. The most common back up mechanism is a tape drive, really just a specialized version of a cassette tape recorder or VCR. These back up drives store data on magnetic tape cartridges like the one shown in Figure 2. As time elapses, the back up media may deteriorate, be discarded or re-used, such that older offline archival data entirely disappears (except, of course, from the many different places it may exist, in bits and pieces, on other servers and local systems).

When e-mail is online, it's an easy and inexpensive task to duplicate the messages and their attachments in their native form to a discrete file or files and burn those to CD or otherwise transmit the e-mail for review and production. When e-mail is offline, it can be no mean feat to get to it, and the reason why it's challenging and costly has to do with the way computers are backed up. The customary practice for backing up a server is to make a copy of specified files and folders containing data. Sometimes a back up will copy everything, including the operating system software and the date; but, more often, time and cost constraints mean that only the stuff that can't be reloaded from other sources gets copied. Another common practice is to only copy all the data every once and a while (e.g., monthly) and just record changes to the data at more frequent intervals. Let's try an analogy to make this clear.

Figure 2

Understanding Server Back Up, by Analogy

Imagine that all your work was done at your desk and that, to protect that work from being destroyed in a flood or fire, you had your assistant photocopy everything on your desk on the first of each month and store it, unstapled and unorganized, in the trunk of your car. Once a month, when the new copy is made, you move the old set from your trunk to your basement. This practice buys you some peace of mind, but realizing that you still stand to lose as much as a month's worth of work should your office burn on the 30th, you figure you need more frequent back up copy sets. Now, neither you nor your assistant can get much work done if everything on your desk is copied every day, so you come up with a shortcut: copy just the new stuff daily (that is, your latest work and your incoming correspondence). Now, on top of the monthly copy of everything on your desk, you add a daily copy of your latest changes. If the office goes up in smoke, it will take some effort to recreate your desktop, but the need to do that only arises in the event of a catastrophe, and you breathe more easily, confident in the knowledge it can be done.

Similarly, incremental server back ups are periodic and pervasive copies of selected datasets, augmented by more frequent recording of changes. Neither alone is complete, but together they comprise a complete dataset at each back up interval.

Coming back to the desktop analogy, some projects linger from month-to-month. Consequently, each monthly interval copy set is going to contain a lot of the same stuff from the month before. Likewise, a server's back up tapes tend to contain a huge volume of duplicate information, interval-to-interval. To forestall the need to wade through many identical copies of the same message, e-mail restored from server tapes must be de-duplicated or "de-duped" to remove repetitious material before review.

But what may be the biggest hitch in doing discovery from back up media is that offline information on back up media isn't accessible in the same way as is online information still residing on the server. Imagine the difference between trying to locate a particular document on your desk--where you are aided by folders, document trays, labels, sticky notes, locale and context--versus trying to do the same while rummaging through heaps of paper in the trunk of your car. Offline back up information usually must be returned to something resembling its former online environment before you can make much sense of it. That may not be a big deal if the systems where that data used to "live" are still around, but it can be a daunting task indeed if those systems were replaced three years ago. In the world of computers, change is constant, and obsolescence arrived yesterday.

You can't imagine how common it is for companies to diligently create back up tapes without ever testing a single one to see if it actually recorded any data. Even when the back up system works, some companies hang onto the tapes but dispose of all the hardware which can read them. In short, never underestimate the power of stupidity. Another point about data stored on tapes: it's fragile. For a host of reasons ranging from sloppy storage to bad hardware to physical deterioration, the usable data that can be successfully restored from a server tape is often less than 100%, and the percentage declines with the passage of time.

Each organization establishes at its own back up practices. Some take the server offline, halting file and e-mail access, while they copy everything. More commonly, incremental back up procedures are employed and may exclude back up of static data, like the server operating system software or packaged commercial applications that can be restored from the original product disks. All Exchange Server back up systems must, over some interval best-suited to the business environment, capture all dynamic data, including:

- System State, including the Microsoft Internet Information Services (IIS) metabase and the Registry;
- Web Storage System (WSS) databases and supporting files;
- Active Directory;
- Key Management Service (KMS) databases;
- Site Replication Service (SRS) databases; and
- Cluster quorum.

If you have no idea what this stuff means, join the club. I'm pretty fuzzy on some of it myself. But, unless you're the system administrator charged with protecting the data, all you may need to know is that back up procedures vary, but they are all geared toward hanging on to the mission critical data.

Brick Level Back Up

By all contemporary standards, e-mail is mission critical data. It's so critical, in fact, that system administrators may elect to back it up in two ways: the global back up touched on above and a mailbox- and message-level approach, commonly called "brick level" back up. If the party responding in discovery maintains a brick level back up system, it's easier and less costly to recover the e-mail of any particular user without having to restore the entire system structure to a recovery server (a spare server used as a target location for recovery operations). With a brick level back up, the system administrator can restore just a single employee's mailbox or an entire department's mailboxes, spitting them out as, e.g., Outlook .pst files for review in e-mail client software or ported into other applications for de-duplication and examination. That's the good news. The bad news is that not every enterprise runs brick level back ups because they take a whole lot longer to complete and use storage space less efficiently than their global counterparts. The lesson may be that, if your litigation needs dictate frequent access to the contents of individual mailboxes stored offline on server back up systems, a brick level back up strategy is best. Of course, if you're getting sued a lot and your opponents are seeking e-mail on your server back up tapes, you've got to also evaluate the strategic implications of making that a fairly easy, less-costly process. Recent trends in electronic discovery cost shifting suggest that getting everything you can into a relatively *in*accessible format may be advantageous to entities resisting discovery.

The Format Fight

Assuming that you've run the gauntlet and gathered all the e-mail files and databases, how are you going to review the fruits of your harvest for relevance, privilege and confidentiality? For any significant volume of data, printing it out and poring through stacks of paper is a terrible idea. You've got to be able to search the material electronically and to access each e-mail's metadata. Here is where the e-discovery world splits into warring tribes we'll call Natives and Missionaries: the Natives believe that e-mail and other electronic data should be searched and produced in its native format, arguing that it's quicker and less costly. The Missionaries preach the gospel of conversion…of data into images, typically TIFF or PDF files, facilitating review via a web browser-like application. For now, the Missionaries seem to predominate, but not for long. Information has simply moved too far beyond the confines of paper. How can the Missionary Model hope to do justice to spreadsheets with embedded formulae, audio content, animation or complex databases? Inevitably, the Natives will prevail; however, the idea of a universal viewer offering easy access to native data by emulating a wide range of common application software should stand the test of time.

For now, the choice of format is a tactical and financial decision. If you know your opponent will find one or the other format more daunting because, e.g., she lacks software to examine files in their native format, that hurdle may influence your choice…in favor of the *easier* format, no doubt. Likewise, if your firm or law department structure is geared to platoons of associates and paralegals conducting discovery reviews using Internet browser software and doesn't have staff capable of analysis in native format, the TIFF or PDF format may be the best choice.

What Format Do You Want?

If you are the party seeking discovery of e-mail, give some careful thought to the format you want to receive and ask for it in your discovery request. But always keep in mind the adage, "Be careful what you wish for, because you might get it." In deciding what to ask for you need to consider how you work and the structure and volume of the electronic information you seek. If you and your staff are incapable of tackling production in any format other than paper and the universe of electronic documents in your case is small (i.e., under 250,000 pages), then working with page image files or even having those images blown back to paper is a workable strategy. In that instance, just be sure that you obtain printouts of all the metadata for each electronic document. That means full headers for all e-mail, plus be sure that the production method will afford you a mechanism to pair attachments with transmittals and link individual messages to the data paths from which they were retrieved (i.e., whose mailbox was it in and what folder?).

Unless you command a platoon of skilled reviewers—or even if you do—once you get past about 250,000 pages, it just doesn't make sense to manage documents by reading each of them. Using my own e-mail stores as an example, I have nearly three gigabytes of e-mail online which, when printed out, might yield something in excess of 300,000 pages to review. If, on average, you can read through every page in thirty seconds, it's going to take around 2000+ hours to plow through it all. Even if you de-duplicate and cull out all the Viagra ads and Nigerian treasury scams, you're still looking at maybe four months of work…for one person…with one mailbox.

For my money, I want the e-mail produced in its native format--in a .pst file if it's Outlook or Exchange Server mail and as .dbx, files (e.g., Inbox, Sent Items, Deleted Items, etc.) if it comes from Outlook Express. Moreover, I'm going to look very closely at the privilege log to determine what has been removed from the mailbox and what relationship those excisions bear to the timing and content of other messages. I'm also going to seek deleted e-mail, whether by examination of server tapes, through discovery from others or by computer forensics.

Privilege and Confidentiality Considerations

If all the cost and trouble of electronic discovery stemmed only from the challenge to locate and restore e-mail, then improvements in technology and best practices could pretty well make those concerns evaporate. The cost of storage has never been lower and the storage capacity/per dollar is soaring. No, the greatest and growing cost of e-discovery stem from the legal services which must be devoted to the fight for access and the review of information before it can be produced. Plaintiff's counsel's fear of overlooking a smoking gun is nothing compared to defense counsel's fear of having unwittingly produced it! Though reprehensible, it's common for confidential e-mails from counsel and transmittals of sensitive trade secrets to rub shoulders with the electronic greeting cards, organ enlargement solicitations and routine matters that fill our electronic mailboxes. Then, there is the commingling of business and personal communications. If e-mail comes from an employee's spouse or physician, who will cull it from production? How do you produce something you haven't reviewed in detail without risking the waiver of privilege?

Claw Back and Quick Peek Arrangements

The inadvertent production of a privileged document following a diligent review and otherwise timely and careful assertion of privilege is not likely to be seen as a voluntary waiver; however, a broad expansion of that proposition---an emerging approach to e-discovery, called the "claw back" or "quick peek" method—offers a less-certain outcome. In a "claw back" production, documents are produced before or even without a review for privilege, confidentiality, or privacy. Instead, the parties agree—or, in rare instances, the court will order—that the party seeking discovery will be afforded broad access, but that the producing party may assert confidentiality and privilege to any of the materials reviewed. The notion is that the producing party may "claw back" any document it might otherwise have been permitted to withhold from production, without fearing a claim of waiver.

Claw back productions certainly have their appeal: make your opponent wade through everything and only focus on the items they indicate they might wish to use. But, even with an ironclad agreement, there is a greater potential for a producing party to waive a privilege or lose control of a confidential communication. There is also a question whether, in our adversarial system, a lawyer's duties are adequately fulfilled by "punting" the review process to one's opponent.

If a claw back or quick peek production is contemplated, one e-discovery think tank suggests that the Court enter an order that (1) indicates that the court is compelling the manner of production, (2) states such production does not result in an express or implied waiver of any privilege or protection for the produced documents or any other documents, (3) directs that the reviewing party cannot discuss the contents of the documents or take any notes during the review process, (4) permits the reviewing party to select those documents that it believes are relevant to the case, and (5) orders that for each selected document, the producing party either (a) produces the selected document, (b) places the selected document on a privilege log, or (c) places the selected document on a non-responsive log. *The Sedona Principles: Best Practices, Recommendations & Principles for Addressing Electronic Document Discovery, Cmt. 10.d* (Sedona Conference Working Group Series 2007).

Preparing for E-Mail Discovery

If a request for production sought, "Jane Smith's Outlook Express e-mail from her Dell laptop, received or sent between March 23 and 30th 2006 and referencing the *Jones Project* in the subject line," electronic discovery would be a piece of cake! In reality, e-discovery requests rarely improve upon their paper discovery predecessors, with drafters opting instead to trot out the familiar "any and all" demand, while tacking "electronic data compilations" onto the litany of examples offered to define a "document."

A lawyer who appears quite savvy about electronic discovery published the following sample request for e-mail production on his website. Ordinarily, I'd credit the source, but since I'm going to savage a well-intentioned and unselfish effort to put something online to help other lawyers, the better part of valor is to let the publisher remain anonymous.

> *"Produce any and all information related to e-mail, including but not limited to current, backed-up and archived programs, accounts, unified messaging,*

> *server-based e-mail, Web-based e-mail, dial-up e-mail, user names and addresses, domain names and addresses, e-mail messages, attachments, manual and automated mailing lists and mailing list addresses."*

Now, let's translate it into a more-or-less equivalent request for paper documents:

> "Produce any and all information related to documents, including but not limited to the original and copies of any documents ever in your possession. Produce any documents you have at home, in your car or that you used to pack the old kitchen dishes you sold on e-Bay. Don't omit all those old compositions you wrote in the 4th grade that your Mom has stored in her attic or the Playboys you kept from college in that box behind the furnace. Produce your Rolodex, your diary, your Christmas card list and that list of the people who gave you wedding presents (your wife will know where it is). Be sure to include any mail you've ever sent or received, especially those blue envelopes with all the coupons in them and any letters from Ed McMahon indicating that, "You may already be a winner!" Produce any implements related to writing, including any pencils, pads, pens and Post It notes.."
>
> Or more succinctly: "Gimme everything."

Sooner or later, your client will get hit with a request like this--or one that isn't utter nonsense—and the reality of having to marshal and produce e-mail and other electronic records will set in.

The process that allows you to safely navigate the treacherous shoals of e-discovery begins *before* the preservation letter arrives. You need a plan. You need a policy. You need procedures.

According to a 2004 survey by the American Management Association, barely a third of employers had written e-mail retention and deletion policies. Cohasset Associates, a consulting firm specializing in document-based information management, found that 39 percent of organizations have no formal policy regarding e-mail retention. Can this really be true after Enron, Zubulake, Frank Quattrone, Morgan Stanley and all the other high profile e-mail self-immolations making headlines?

Planning and Policymaking

Companies get in trouble with e-discovery because they fail to keep something and create or retain something they shouldn't have. In a large, complex, far-flung organization, it's bound to happen despite best efforts, but it shouldn't occur because the law department doesn't know how to talk to the IT department or because no one ever told Dewayne that he shouldn't e-mail his favorite scenes from "Borat" to the whole department.

Your client's electronic document retention policy has become a critical corporate policy. Having a sound retention policy and implementing it in a rational and consistent way is one of the best ways means of guarding against a charge of spoliation. Such a policy needs to be a

collaborative effort between corporate counsel and the IT staff, with each seeking to understand the needs and constraints the other faces. The policy needs to be blessed by senior management and integrated into operations. It needs to be ingrained in the corporate culture by training, oversight and meaningful enforcement. Currently, most employers don't instruct their employees on proper handling of electronic records, and almost three out of four have no e-mail usage training. A policy without training and enforcement is just a piece of paper.

Dear David Duncan, Regards Nancy Temple

There's been plenty of ink spilled about the demise of accounting giant Arthur Andersen in the Enron mess, but one pertinent lesson is that Andersen didn't get in trouble because it lacked a document retention policy—in fact it had two pretty comprehensive document destruction policies. Andersen went down because it hadn't *followed* its policies and decided to play catch up and cover up while the Feds were pulling into the driveway. Few things spell "wrong" to a jury like a company's failure to adhere to its own policies. Some argue it's better to have no policy than one that's not followed.

To be effective, retention schedules have to be rigorously followed, but adaptable to lawsuits, government investigations and compliance obligations. The retention policy that only kicks into gear when the hoof beats of litigation approach waves the red flag of malfeasance. Yet more than a third of companies only follow their retention when it suits them.

Trust Everyone, but Cut the Cards

Even companies with sound e-mail usage and retention policies and employee training programs can't wholly rely upon their employees' good conduct. Employees must be disabused of the notion that they have an expectation of privacy in their use of company computers and reminded that their usage constitutes consent to monitoring of that usage. Monitoring of computer usage may be degrading and intrusive, but failing to monitor is an abrogation of responsibility that cedes trade secrets to those who steal them and vast digital conduits to those who use them for harassment and criminality. These threats are not imaginary. They occur in every large organization, and many small ones, from the board room to the mail room. Moreover, we must have the fortitude to look for the bad guys, inside and out. Though half of all companies claim to monitor incoming e-mail, less than one-in-five keep an eye on intra-company messaging.

Am I in Trouble? IM!

I used to call Instant Messaging "an emerging threat," but Punxatawney Phil already emerged and saw his shadow. Now we can look forward to six more years of S.E.C. investigations! Seriously, IM is in wide use throughout corporate America. Estimates of office usage range from 45%-90%, with an expectation that, whatever the real usage, it's getting bigger all the time. For the uninitiated, IM is a form of instantaneous, real time e-mail that doesn't come though normal e-mail channels, meaning it's largely invisible to those whose job it is to police such things. IM leaves little in the way of digital footprints, which may be desirable if you're using it to play footsie on company time; however, unmonitored and unrecorded communications pose an entirely different risk to financial institutions. For example, the

National Association of Securities Dealers (NASD) requires members to archive electronic communications for at least three years. NASD Chairman Mary L. Schapiro said, "Firms have to remember that regardless of the informality of instant messaging, it is still subject to the same requirements as e-mail communications and members must ensure that their use of instant messaging is consistent with their basic supervisory and record keeping obligations." So, how come 61% of financial services firms surveyed by Security Industry News have no means of managing or archiving instant messaging, and 39% have no instant messaging policy at all? I forget, when the law says you must retain it and you don't, is that spoliation *per se* or just a felony?

> **Solution**: Firms must either bar IM usage altogether and monitor the Internet ports used by such applications to insure compliance, or allow such usage configured so as to permit monitoring and archival. Doing so won't be a one-time fix, because IM applications evolve rapidly, such that a message going out one port today will bust through the firewall an entirely new way tomorrow.

Training

I have an idea that might protect a company from employee e-mail gaffes. It involves putting a giant video screen in the company cafeteria and randomly displaying the contents of any e-mail going through the network. It's a nutty idea, but it makes the point: Before they click on "Send," every employee needs to ask, "How would I feel if I had to read this in open court or if my kids heard it on the evening news?" Sensitivity to the perils of e-mail doesn't just happen—it has to be bred institutionally, and it needs to come from the people at the top and matter to the folks at the bottom. In 1945, people understood that, "Loose lips sink ships." In 2007, every employee needs to feel—and every co-worker should serve to remind them—that an inappropriate, illegal, misdirected or mishandled e-mail puts everyone's livelihood at risk.

> **Solution:** Just reminding employees that the company has an e-mail policy is not enough. There must be formal training on appropriate content. Retention policies must be spelled out, and employees should be made to understand why compliance matters—that when you don't do what the policy requires, you're betraying your co-workers. Teaching ways to avoid misdirection (e.g., turning off the auto complete feature for addressing e-mails) and encouraging the same level of reflection attendant to a written memorandum will help.

Social Engineering

Social Engineering is hacker-speak for tricking a person into revealing their password or launching a rogue program to open a back door into a system. I use it here to underscore the fact that the weakest security link in most systems isn't the software or the hardware. It's the "wetware," also called "liveware" or "meatware." That is, it's the people. The best planned systems are waylaid by the people that use them.

By way of example, since more than a third of companies store their e-mail solely on servers, system administrators are forced to limit mailbox size. In fact, three-fourths of companies surveyed by Kroll Ontrack impose such quotas, and a quarter of companies compel deletion

as quotas are reached. When you tell employees that you are going to force them to delete what many view as essential information, not surprisingly some become quite resourceful at retaining e-mail despite company policy. Avoidance tactics take many forms, but whether it's forwarding older mail back to your own mailbox to circumvent time restrictions or burning private caches of CDs, such guerilla tactics jeopardize a company's ability to manage their e-mail systems and accurately respond to discovery. That's bad social engineering. An enterprise embroiled in litigation may vehemently deny the existence of responsive e-mail, only to find that an enterprising employee has a "private stash" of clearly-discoverable e-mail which does not come to light until the employee deems disclosure of that e-mail advantageous. As attorney Tom Watkins of Austin puts it, "E-mails are the cockroaches of litigation. You can't get rid of them, and they always manage to turn up when company comes to call."

> **Solution**: Build institutional awareness of the hazards of kamikaze computing. Train, monitor, audit and enforce. People try to get away with stuff because they can. Make it harder to cheat, and put real teeth in the policy. Help employees appreciate the risk to their company and their jobs posed by social engineering errors, and put peer pressure to work.

The E-Discovery Triage Plan

One of the earliest obligations of any litigant is to preserve evidence in anticipation of litigation. The duty to preserve is automatic, and doesn't hinge on suit being filed or even receipt of a preservation letter. Companies have to be prepared to retain evidence when litigation or government investigation is merely "in the wind." The role of harbinger often falls to corporate counsel, who must issue something of a "stop the presses" order to be sure that appropriate steps begin at once to preserve potential evidence.

If it fell to you to initiate the preservation of potential electronic evidence, would you know what to do? Would you even know everyone that must become involved? Would the IT department understand what they were required to do and have the resources and in-house expertise to do it?

If you're at all uncertain of your answers to the prior questions, you may need an e-discovery triage plan—the procedural equivalent of a big red button in your office you can push when you need to "stop the presses." An e-mail triage plan starts with knowing the systems and staying current on the nature and location of the servers, back up archives and other key data repositories. It requires having at hand the names and contact information for the persons in each department who have the authority and knowledge to preserve and protect potential evidence. It means knowing where the e-mail lives on the company's systems and halting activities that might destroy or alter those messages.

An e-mail triage plan needs to keep close tabs on all potentially significant sources of discoverable information. Who telecommutes and may have electronic evidence on a local hard drive in their home? Who's been issued a company-owned laptop, Blackberry or PDA that might hold e-mail or other evidence? How often is the e-mail server backed up? How complete is that back up? Do we need to temporarily implement brick level back ups? What

is the rotation schedule for the back up tapes? What local hard drives need to be cloned immediately? What about Instant Messaging and voice mail?

Electronic data is fragile, and the cost of spoliation is high. To best serve your clients, you should stay abreast of how their IT systems retain electronic documents, and, if necessary, propose changes in procedures to support an e-discovery triage policy. The point at which the duty to preserve attaches is not the time to begin your education about the company's systems or start seeking management buy-in on a preservation plan. You must be fully prepared to preserve the status quo, to—as far as feasible—fix the company's data in amber for the near term, long enough to secure agreements with opposing counsel or relief from the court. The moment the duty to preserve attaches is likewise not the time to engage in a power struggle with the IT department. Make it your business to know who you will be dealing with and meet them. Discuss the e-discovery triage plan and inquire about potential conflicts or concerns. Though such a plan should have emerged as a collaborative effort, it's still a good idea to secure buy-in and solicit ways to improve the plan. In short, *communicate*.

Tips for your E-Discovery Triage Efforts:

1. Field an E-Discovery Triage Task Force and include:
    a. Corporate Counsel
    b. Outside Trial Counsel
    c. IT Officer(s)
    d. Records Custodian(s)
    e. Chief Financial Officer
    f. Operations Officer
    g. Electronic Discovery Specialist
    h. Forensic Specialist

2. Define the product of the Task Force: Are they drafting the company retention policy or a litigation action plan? What is each member's role and responsibility?

3. Identify all data storage locations and a mechanism to stay abreast of changes

4. Document existing procedures and schedules for creation, storage, retention, modification, securing, deletion and restoration of business data;

5. Identify likely candidates for discovery efforts and effective ways to delineate or "Chinese Wall" privileged, personal and confidential data, as well as to retain and retrieve.

6. Develop action plan procedures for particular events including employee departure, suspected theft of trade secrets, network intrusion, improper or unauthorized use of computer systems, government subpoena, FBI raid, employee destruction of data, litigation, etc.

7. Create a contact list of persons responsible for familiarity with and implementation of the action plan and insure a rapid and effective communication strategy. How will everyone "get the word" to act?

8. Secure support from top management to insure prioritization and avoid delay in implementation.

Enlist Help

Even with a well-conceived e-discovery triage plan, it's a good idea to get outside help on board to advise and assist. Why would a big communications or technology company need to bring in outside help to assist with electronic discovery? Do Microsoft, Google or Dell really need outside expertise? The answer is often "yes;" not because an outsider necessarily brings more knowledge of the systems or mastery of the technology, but because a well-chosen outsider brings an independent voice to the table and speaks the language of the IT department at a time when clear communication is essential. Despite being paid by a party, an expert known to the court and enjoying a reputation for honesty and skill is simply more credible when stating, "We looked and it wasn't there," or "The items reviewed were not responsive to the request." Moreover, hiring outside talent helps demonstrate that discovery responsibilities were taken seriously and—let's be blunt here—it may serve to deflect responsibility if something should ultimately hit the fan.

Control the Channel and Capture the Traffic

As a forensic examiner, I've see that a common consequence of telling an employee that someone will stop by tomorrow to pick up their laptop is that they will be up most of the night running a Delete-O-Thon. Then, a case which might have been won is lost; not on the merits, but because of a failure to control the data channels and capture the traffic. You must be able to lock down your records into a full save mode upon the hint of litigation or investigation. You need to make users aware that not only must they keep their personal and sexual material off their company computers else they be content to hand it over when the time comes. Clients need to appreciate that those "evidence eliminator" programs that promise to cover their tracks don't do a very good job of it. Plus, covered tracks on a computer look just like—surprise!—covered tracks. Even if I don't find the erased item, chances are I'm going to find the crater it left behind.

"Controlling the channel" demands more than an occasional e-mail admonishment to "hang onto stuff." The average user has, at best, a hazy idea about how computers keep and lose information. You need to be explicit about what must be done or not done on desktop and laptop systems, and do it in such a way that it won't appear as a roadmap for running that Delete-O-Thon!

Consider hardware and software "solutions" that enable more centralized control of the retention process. Some of these will even image hard drives remotely to permit a "snapshot" to be taken of each user's hard drive during off hours. If it sounds a bit Big Brother, it is. But better Big Brother than Brother, can you spare a dime?

The Server Tape Conundrum

According to the market research firm Osterman Research, 67 percent of companies back up their e-mail systems to tape alone and recycle the tapes every 90 days. Suppose you know that your client's server data are backed up to tape and that those tapes tend to be re-used in a way that overwrites old data with new. When the time comes to swing into action and preserve potentially discoverable evidence, how are you going to deal with your client's tape rotation? The easy answer is, "I'll instruct them to stop re-using tapes until further notice." That's certainly not a wrong answer from the standpoint of protecting your client from claims of spoliation and even from the Delete-O-Thon initiatives of their own employees, but it's not always a practical or tactically sound one. It's the right answer according to 7 Moore's Federal Practice, which states that, "The routine recycling of magnetic tapes that may contain relevant evidence should be immediately halted on commencement of litigation." § 37A.12[5][e] (Matthew Bender 3d ed). But, it is not the *only* right answer, nor is it necessarily the right answer from beginning to end of the litigation.

Many companies are *always* embroiled in some phase of litigation, so an instruction to cease back up rotation during the pendency of a case is tantamount to saying, "Save everything forever." Back up tapes are expensive. Properly storing back up tapes is expensive. Hanging on to the obsolete hardware needed to read back up tapes from last year or the year before that is expensive. There are specialists who make a handsome living curating "Museums of Old Backup Tape Drives" because no one thought to hang onto those tape drives from 1995 or the software than ran them. Even when the case from 2004 is finally over, do you have to retain the tapes because of the case filed last month?

What you advise your client to do and for how long should be based in part upon how they use their back up system. Companies tend to fall into two camps: those that use their back up systems as a means to recover from catastrophe—to get their systems "back up" and running again—and those that use back up as a means of institution memory--an archives of company activities extending beyond the minimum required to restore to the point of failure. If your client falls in the latter camp, they almost certainly ***do*** need to halt their tape rotation, since their usage is archival of business records and the start of litigation is an inauspicious time to start destroying business records, at least until you can fully ascertain the relevant scope of the matters in dispute. But if your client falls in the first camp and just uses back up to get back up, doesn't maintain an archive of old tapes and keeps the focus solely on catastrophic recovery, you may be fully justified in not halting back up tape rotation, assuming that you have taken other appropriate steps to preserve potentially relevant and discoverable data. Keep in mind that, absent a catastrophic failure, the most recent back up set is essentially a mirror image of the live system data, so restoring and searching the latest back up is usually of little value.

Before you decide in which camp your client falls, you'll need to do more than just ask the V.P. of IT whether there is a tape archive. You need to pose your questions as well to the person whose job it is to shove those tapes into the machine and keep track of them. The reality is that the manager may not always know what the technicians are doing "in the pits."

If you take the safe route and order a halt to rotation of back up tapes, recognize that there are costly consequences which follow upon that instruction and promptly explore whether

there are less-costly alternatives. Perhaps the court will enter a discovery order making it clear that back up tapes need not be retained or an agreement can be reached with opposing counsel to the same end. A motion seeking cost allocation for back up tape retention costs can sharpen opposing counsel's focus on the issue. As plaintiff's counsel, I know I was very careful about what I sought in discovery when I thought it might come out of my pocket. Also, target follow up dates to advise IT about the need for continued retention. It would be embarrassing to find out that IT unnecessarily spent $22,000.00 this year on litigation-related back up activities because you forgot to tell them the case settled last year!

Confer, Confer, Confer!

Voluntarily sharing information with your opponent and seeking to work cooperatively in the electronic discovery process may not be your cup of tea, but it's certainly an effective way to protect your client from claims of spoliation and discovery abuse. Huge sums are spent on electronic discovery because of uncertainty—we're not sure what we must keep, so we keep everything.

The better way is to confer with your opponent early. Document the process well and seek to hammer out a discovery plan setting out what you are agreeing to preserve pending specific discovery requests. Be prepared to ascribe estimated volume and costs to more extensive retention efforts so that your opponent appreciates the costs occasioned by overbroad demands. Such a conference is less about agreeing to produce particular items as it is defining the universe of information to which future discovery will be directed. Why should your opponent agree to limit that universe and cede a tactical advantage? Because, if you've made your case that your opponent's demands are unreasonable and put your opponent on notice that money will be wasted as a consequence, you are better postured to shift that financial burden to the other side, or at least have it dangle over your opponent like the sword of Damocles.

The other reason to confer and seek agreements early is because limiting electronic discovery is a two-way street. Many discovery requests can be "boomeranged" back to your opponent, who will be hard-pressed to object to its scope. A common error of corporate counsel is to think that the cost, complexity and peril of electronic discovery are visited only on their side. Nearly everyone uses computers. Though the party litigating against your corporate client is an individual, they are likewise bound to preserve electronic evidence, a treacherous and costly obligation for the uninitiated, even for a single personal computer. A conference—and incisive questions about what steps the other side is taking to preserve evidence—may bring the parties closer to agreement.

If agreements can't be reached, seek a discovery conference with the court and help the judge appreciate the costs and perils of willy-nilly retention. Be prepared to discuss volumes of data, man-hours of work and associated costs. Few judges respond favorable to a plaintive, "It's too burdensome," but most, when made aware of the dollars and time at stake, are willing to use their power to prevent unfairness and waste. Help the court see alternatives—sampling, perhaps, or time limitations—to a global retention obligation. Even if you get no relief at all, you can better advise your client that the money and time being

invested is indeed required, and you set the stage for a later cost allocation request should it appear that your opponent overreached or oversold.

Twenty Tips for Counsel Seeking Discovery

1. Get your preservation letter out early and be *both* specific and general. Assume that the recipients don't know their own systems and don't understand computer forensics. Educate them in the letter so they can't use ignorance as an excuse.
2. Do your homework: use the Net and ask around to learn about the nature and extent of your opponent's systems and practices. You're probably not the first person to ever pursue discovery against the opposition. Others might know where the sweet spots can be found.
3. Get your e-discovery out fast, with the petition if you're the plaintiff. Data is going to disappear. You're in a poor position to complain about it if you didn't ask while it was still around.
4. Force broad retention, but pursue narrow discovery
5. What they must preserve and what they must produce are very different obligations. Keeping the first broad protects your client's interests and exposes the opposition's negligence and perfidy. Keeping requests for production narrow and carefully crafted makes it hard for your opponent to buy delays through objection. Laser-like requests mean that your opponents must search with a spoon instead of a backhoe. Tactically, ten single, surgical requests spread over 20 days are more effective than 20 requests in one.
6. Be aware that your opponent may not understand the systems as well as you do, but may not want anyone—especially his client--to know it. Help your opponent "get it," so he can pose the right questions to his client.
7. Question the IT people. Avoid the managers and focus on the grunts. The latter are have spent less time in the woodshed and they know the *real* retention practices.
8. Seek a copy of any document retention policies and a complete inventory and topology of system resources. You need to know where the data is stored and on what equipment.
9. Invoke the court's injunctive power early to force preservation. The agreement that can be secured to forestall a court order may be better than you'll get from the judge.
10. If you can't get make any headway, seek appointment of a neutral or special master.
11. Ask all opponent employee witnesses what they were told to do in the way of e-document retention and what they actually did.
12. Know how and when to check for authenticity of data produced. Digital data is easily forged.
13. Be sure to get metadata whenever it may be relevant.
14. Don't accept image data (TIFF or PDF) when you need native data.
15. Have the principal cases and rules on e-discovery and cost shifting at hand. Tailor your requests to the language of the cases and the rules.
16. Set objections for hearing immediately. Require assertions of burden and cost to be supported by evidence.
17. Analyze what you get promptly after you get it and pin down that it is represented to be "everything" responsive to the request. Follow up with additional requests based upon your analysis.
18. Don't let yourself be railroaded into cost sharing but, if it happens, be sure you're protected from waste and excess by the other side, and leverage your role as underwriter to gain greater access.

19. Be prepared to propose a "claw back" production, if advantageous.
20. Don't accept assertions of cost or complexity unless you know them to be accurate. Have such claims independently evaluated and be ready to propose alternatives.

Twenty Tips for Counsel Defending Against E-Discovery

1. Respond immediately to any preservation letter and advise what you will and won't do without a court order and why. Don't enable your opponent to later claim, "I thought they were saving everything I asked for."
2. Act immediately to preserve potentially relevant data. Know the tape rotation schedule and decide whether to halt rotation and to what extent. Communicate clearly and specifically what your client's employees must do and for how long. Don't rely on intermediaries if data destruction is in the offing. You may only get one shot to preserve some things, so don't just leave a voice mail for someone who's away on vacation. Implement your e-discovery triage plan, and be sure that management gets behind it unequivocally.
3. Confer with opposing counsel early and often. Document everything you proposed, agreed or declined to do.
4. Seek a discovery conference with the court if the retention or production obligations are onerous.
5. Meet with the IT staff and let them help you understand what must be done to respond to a request and whether it can be done. Have them propose alternatives. Treat them as "officers of the court" within their digital domain.
6. Prepare IT staff and records custodians for deposition--not just the department head. Be sure they know the retention policy and how it has been implemented. Engineering types tend to look for solutions, so caution them against helping your opponent solve her problem of getting what she seeks from your systems!
7. "Boomerang" your opponent's discovery where advantageous, serving it back on the other side. More importantly, ***push back with e-discovery***. Responding to an electronic discovery request is a perilous undertaking even when you only have one computer (most Americans have more than one). Even if you are Goliath, the David suing you doesn't have an IT staff and may be unable to resist the temptation to sanitize his e-production. "David" may be no more inclined to share his e-mail with you than you with him. Moreover, home computers tend to reveal much more than their office counterparts, so consider computer forensics as well.
8. Document all efforts to identify responsive material. Should something be missed and you need to show good faith, it will take more than a global representation of, "We looked really hard." Quantify efforts in bankers' boxes, gigabytes, man-hours and hard dollars. This information will also be useful when seeking to demonstrate the burden imposed by future requests and when seeking to shift costs.
9. When appropriate, seek to shift costs to your opponent. A credible risk of paying your client's bills is a very big hammer, but be sure that the Court doesn't confuse cost shifting with broader access. Just because the opponent has to pay for the collection and search effort doesn't confer a greater right to see anything.
10. When claiming undue burden, be prepared to attach reasonable estimates of time and money to responsive efforts. Be sure the court understands that employee time isn't "free." Get quotes from outside vendors to support credibility. Don't forget the cost of

review by counsel. It may not be shifted, but it is a major cost consideration capable of making an impression on the court. Help the court appreciate that a discovery request that costs you more than the settlement demand is a tactical ploy that doesn't serve the ends of justice.

11. Know the new Federal e-discovery rules and the seven Zubulake v. UBS Warburg LLC, 217 F.R.D. 309 (S.D.N.Y. 2003) cost shifting considerations, and be ready to apply them:
    a. *Is the request specifically tailored to discover relevant information?*
    b. *Is the information available from other sources?*
    c. *How does cost of production compare to the amount in controversy?*
    d. *What are the relative positions of the parties in terms of resources?*
    e. *Who is best able to control costs and has an incentive to do so?*
    f. *Are the issues in discovery key to the issues at stake in the litigation?*
    g. *What are the relative benefits to the parties of obtaining the data?*
12. Consider sampling, filtering and targeted search as alternatives to broad production.
13. Be sensitive to undisclosed concerns stemming from private information on hard drives which may cloud judgment. The CEO may know that he has a porno collection hidden away on his office computer, but he's unlikely to admit it to counsel.
14. Be wary of forensic analysis of hard drives by the other side's expert. Almost everyone has something to hide, and a lot of them hide it on their computers.
15. A back up is just for getting the system "back up" after a crash. If your client doesn't need old back up tapes to get back up, then get rid of them! Keeping them tends to makes them discoverable as a business record. Being a digital pack rat is what gets so many companies into costly hot water.
16. Educate yourself about computer system and storage, so you can educate the court.
17. Protect your client by protecting the interests of third-parties. Raise claims of third-party privacy and privilege rights where such claims are genuine, material and will serve as grounds for non-production. Office e-mail oftentimes contains privileged attorney-client and spousal communications as well as confidential medical information. Complying with discovery may expose you to liability to third-parties.
18. Anticipate leaks in the net: Retired hardware, crashed drives, and employee pack rats are all places where you may find data all swear is gone forever. Look in closets, drawers and on shelves!
19. Systematic retrieval starts with the sender. Encourage clients to train employees to use e-mail properly, label subject lines accurately and avoid threading.
20. Make sure your clients appreciate that failing to produce unfavorable electronic evidence—especially the smoking gun e-mail—is an invitation to disaster. You can't suppress all copies, and you can't be sure the other side won't get it from somewhere else. It a*lways* hurts more when it's introduced as something you tried to hide.

Finding the Right Computer Forensic Expert

Craig Ball

Finding the Right Computer Forensic Expert
Craig Ball

Is deleted-but-not-gone electronic evidence a "bet the case" concern? Ask Morgan Stanley or domestic maven Martha Stewart or convicted murderer Scott Peterson. Ask anyone at accounting giant Arthur Andersen. Wait, can't do that. Arthur Andersen is *gone,* hoisted on a petard of e-mail and shredded work papers.

Far more information is retained by a computer than most people realize. You could say that a personal computer operating system never intentionally erases anything, even when a user deletes a file. Instead, PCs just hide a deleted file's contents from view, like crumbs swept under a rug. Computer forensics (CF) is the identification, preservation, extraction, interpretation and presentation of computer-related evidence. It's reconstructing the cookie from the crumbs. But unless specialized tools and techniques are used to preserve, examine and extract data, and proper interpretive skills are brought to bear, evidence will be lost, overlooked or misinterpreted.

Everyone uses computers. If you're a prosecutor, litigator or in-house counsel, a computer forensics expert is in your future. You *must* know how to choose a CF pro for your side or test the opposition's choice.

Computer forensic examiners aren't licensed as such. No "bar exam" establishes their competency. Anyone can put "computer forensic examiner" on their business card. Nevertheless, a cadre of formidably skilled and principled computer forensics examiners remains the core of the profession. The challenge is to tell one from the other and to help the judge and jury see the difference, too.

Finding a CF Expert

The best ways to find a good CF expert are the same used to find experts in any technical discipline: ask other lawyers and judges who to use and avoid, and delve into the professional literature to spot scholarship and leadership. If you practice in a small community and can't secure local recommendations, contact one of the professional associations for CF examiners (the High Technology Crime Investigation Association at www.HTCIA.org is the largest) and get the names of nearby members. Internet searches for experts may turn up worthwhile leads, but don't judge qualifications by where the expert appears in a search engine. It's just too easy to buy or engineer favorable placement. Instead, use the 'net to troll for publications and for networking. The non-commercial Electronic Evidence Information Center (www.e-evidence.info) is a superb starting point for a wealth of information on leading computer forensics practitioners.

Many experienced CF examiners come from law enforcement and the military. Look for, e.g., former DOD, IRS, FBI and Secret Service credentials. Sadly, child pornography represents the bulk of CF work by many ex-law enforcement investigators, so ask about broader experience with other computer crimes. Extensive experience on the civil side is a plus.

Plenty of computer savvy folks lacking forensic training or experience offer their services as experts. But, just as few doctors are qualified as coroners, few systems administrator have any forensic qualifications. A background in law, law enforcement or investigation is important, whereas programming experience has little bearing on computer forensic ability. Be certain to obtain the witness' C.V. and check it for accuracy. Look for membership in professional CF associations, formal training and certification. Has the expert published articles on computer forensics or regularly participated in public online CF forums? Read these contributions to gauge knowledge, commitment to the profession and communication skill, then weigh the following when evaluating qualifications:

Is the examiner certified?
An increasing number of organizations offer certification in computer forensics. Some, like CCE and ENCE, indicate real expertise and others mean little. In evaluating certification, find out exactly what the expert had to do to be certified. Was there a background investigation? Was written testing required? Was there a practical component? What about peer review and a minimum experience threshold? Who taught and certified the expert? Do any applicants fail to obtain the certification? Was expertise certified in a discipline or in the use of a particular tool or software package?

How much time devoted to computer forensics?
Question the focus of a CF expert wearing many hats for hire as, e.g., PC repair specialist, network installer, programmer or private investigator. A large firm's far-ranging claims of expertise may be justified, but for the solo or small shop expert, "dabbling" in computer forensics is not an option.

How experienced as a witness?
If the expert you're evaluating held up in past courthouse challenges, chances are she will again. Look for experience in the type of case you're handling. A veteran of porno prosecutions may not be well suited to a case involving employment discrimination or IP theft. You can't be an fully effective CF examiner if you don't understand what the case is about, so be certain your choice knows the ins-and-outs of civil litigation. Talented CF experts convey hyper technical concepts without lapsing into jargon or acronyms and possess easy facility with simple analogies.

How much classroom training?
Ideally, a CF expert has been formally trained and can demonstrate dozens or hundreds of hours of CF classroom work. Note, however, that some of the best qualified experts in computer forensics have little or no formal training in the discipline. They're largely self-taught and have been at it since the dawn of MS-DOS. These veterans, too, should be able to demonstrate time in the classroom…as the instructor.

What will it cost?
Good computer forensics is expensive. Even a basic computer forensic examination costs several thousand dollars or more. A complex exam can run to six figures. One veteran examiner analogizes that a top-notch cardiac surgeon can teach anyone to perform a routine

heart bypass in an afternoon—it's just plumbing—but the necessary expertise and attendant high cost spring from the decades it took to learn what to do *when things go wrong.*

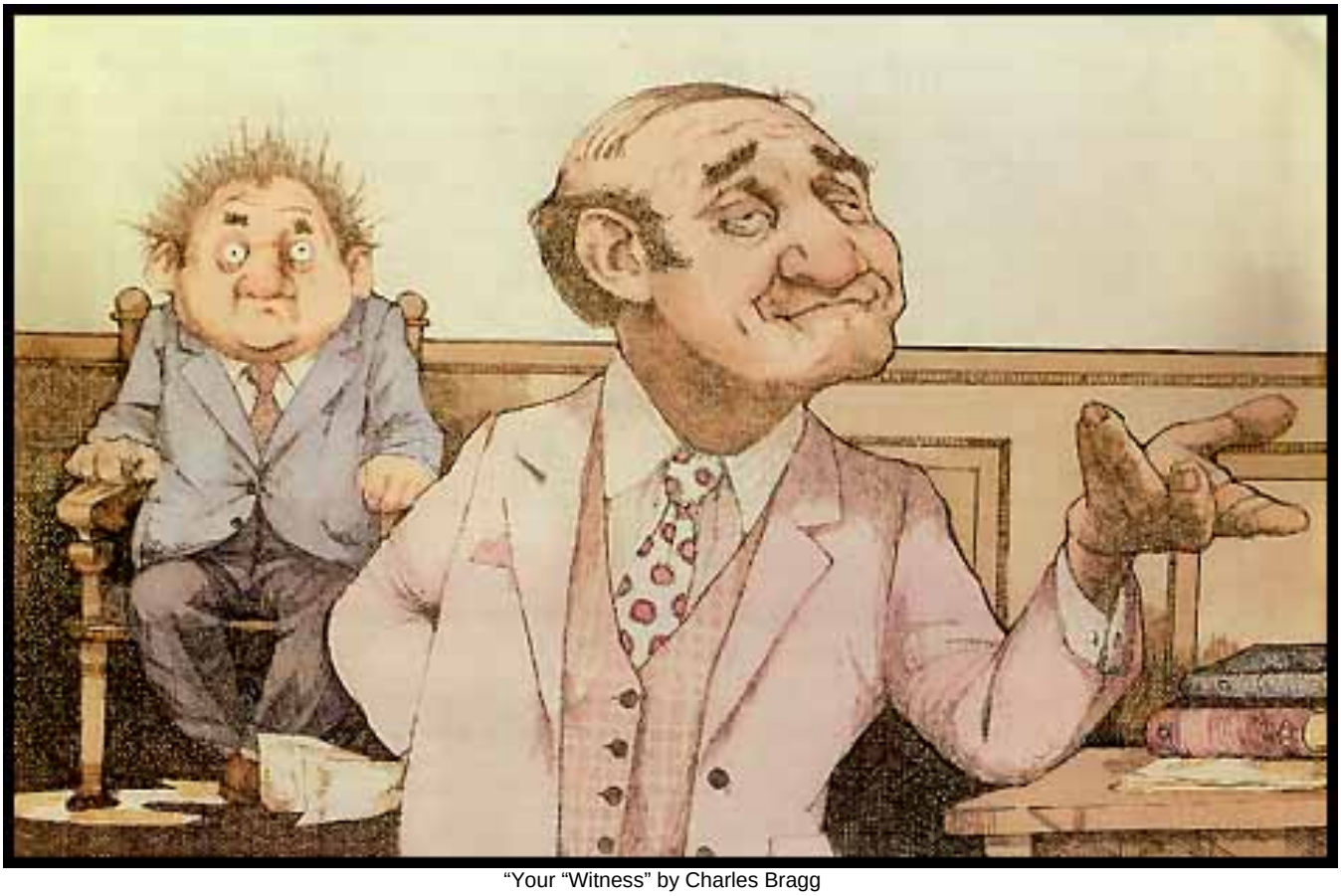
A CF expert should clearly communicate hourly rates and anticipated expenses, but there are typically too many variables to quote a bottom line cost. If you can supply reliable information about the systems, electronic media and issues, experience may permit the expert to project a range of expected cost. Recognize that competent examiners routinely decline requests for a "two-hour quick peek." No one wants to be taken to task in court for missing something because they didn't have time to do the job correctly.

What do other clients think?

Before you commit to spend thousands, ask for references and spend a few minutes calling attorneys who've worked with the expert. Some client identities might be withheld as confidential, and those supplied probably won't be the disgruntled folks, but you're sure to glean *something* useful respecting billing practices, reporting skill, discretion, preparation or professionalism. If nothing else, an expert unable to identify satisfied clients might not be the one for you.

Beware of the Tool Tyke

Poorly-trained experts rely on software tools without understanding how they work. They're Tool Tykes. Of course, all of us trust technologies we don't fully understand, but an expert should be able to explain *how* a tool performs its magic, not offer it up as a black box oracle. Tool Tykes dodge attacks on their lack of fundamental skills by responding, "The tool is not on trial," or citing how frequently the testimony of *other* witnesses using the same tool has been accepted as evidence in other courts. The use of proven tools and software is essential, but even a rock-solid tool in unskilled hands is unreliable. Forensic software suites are principally designed to automate repetitive tasks that would otherwise be performed manually. Your expert should understand those underlying operations, not just know the keystroke required to initiate them.



"Your "Witness" by Charles Bragg

Cross-examination of the Computer Forensics Expert

Craig Ball

Today, some 95% of all documents are created using computers. Daily electronic mail traffic far outstrips postal mail and telephone usage *combined*. Computer technology impacts every facet of modern life, and the crimes, torts and disputes which carry us to the courthouse are no exception. The new field of computer forensics entails the identification, preservation, extraction, interpretation and presentation of computer-related evidence. Far more information is retained by a computer than most people realize, and without using the right tools and techniques to preserve, examine and extract data, you run the risk of losing something important, rendering what you find inadmissible, or even causing spoliation of evidence.

Though I've been immersed in computer forensics as a trial lawyer and as a computer forensics student, examiner, author and instructor for many years, I'd never come across an article that offered practical advice on the cross-examination of a computer forensics expert. The goal of this paper is to improve the caliber and candor of those who testify as computer forensics experts and to help lawyers get to the truth, not confuse or obscure it.

The Cops-and-Robbers Mindset

The world of computer forensics is heavily populated by former law enforcement officers from the Secret Service, FBI, Treasury, military investigative offices and local police forces. Many of these veteran officers--though generally well trained and very capable--retain a good guy/bad guy mentality and some regard computer forensics as a secret society where they don't want the "bad guys" to know their secrets. Lawyers are seen as aiding the bad guys, and the very last thing forensic examiners want is for lawyers to understand the process well enough to conduct an effective cross examination. With some justification, former cops view lawyers with suspicion and even disdain (how this makes them different from the rest of the world, I don't know). To their way of thinking, lawyers are contemptuous of the truth and bent on sowing the seeds of distraction, confusion and doubt.

This mindset can make forensic examiners guarded witnesses: not necessarily hostile, but reluctant, or quick to dive under cover of technical arcana and jargon to shake off a pursuer. A forensic examiner is dealing with largely objective observations and shouldn't come across as an advocate. If evasive or uncooperative on cross, give the witness enough rope for the jury to see it.

Tool Tykes

Poorly trained experts rely on software tools without fully understanding how they work. They're Tool Tykes. Of course, all of us trust and swear by tools we don't fully understand--do you really fathom how a quartz wristwatch tells time or a mouse moves the cursor?—but, an expert should be able to explain *how* a tool performs its magic, not offer it up as a black box oracle. Tool Tykes are trained to dodge attacks on their lack of fundamental skills by responding that, "The tool is not on trial" or citing how frequently the testimony of *other* witnesses using the same tool has been accepted as evidence in other courts. Don't let them get away with this evasion. A great tool in unskilled hands is not reliable. Press the witness to either explain how the tool achieves its results or admit they don't know. Be advised that this technique will flush out only the pretenders to the throne of "expert." Real pros are going to know how their tools work down at the bit level and be able to explain it in a way any juror can grasp. Of course, *you* should be ready to distinguish the right explanation from technical doubletalk.

Computer forensics is a new discipline and many computer savvy persons without forensic training or experience offer their services as experts. Just as not every doctor is qualified as a coroner, not every systems administrator is a forensics expert. A background in law, law enforcement or investigation is important, whereas programming skills have little bearing on computer forensic skills. Be certain to obtain the witness' C.V. and check it for accuracy. Look for membership in professional associations of computer forensic examiners, formal training and certification. Find out if the witness has published articles on computer forensics or participated in public list serves supporting the discipline, then find and read those contributions to assess their expertise.

Chain-of-Custody Issues

Because of their law enforcement backgrounds, forensic experts tend to be very savvy about the importance of, and the proper procedures to maintain, a chain of custody. A chain of

custody attack is warranted when you can level a credible charge that someone tampered with the evidence. The critical importance of the chain of custody is drilled into every computer forensic expert. If you can prove the witness botched the chain of custody, the witness will be shaken and defensive. Even when tampering isn't suspected, a sloppy chain of custody suggests a poorly qualified expert.

The Limits of Computer Forensics

Nearly everyone uses computers, but few users understand them well. A witness who's mastered the computer's deepest secrets may enjoy a Guru-like authority when testifying. If you're seeking to cast doubt on the witness or the science of computer forensics, you may gain traction by getting the witness to concede some of the things an examiner *can't* ascertain about how a particular computer was used or who used it.

Though computer forensics specialists can perform miraculous tasks, there are limits to what we can divine or resurrect. Some of these limits are oddly mundane. For example, it can be difficult to establish that a user altered the time on their computer, especially if the clock has been correctly reset and logs adjusted before the examiner arrives. Computers are pretty "stupid" where time is concerned. A toddler (at least one who doesn't live in Alaska) would challenge the assertion that it's midnight if the sun's still up, but, no matter what the actual time may be, a computer accepts any setting you give it as gospel. There are ways to ferret out time manipulation, but they aren't foolproof.

Similarly, a computer can't identify its user. At best, it can reveal that the user was someone with physical access to the machine or who perhaps knew a password, but it can't put a particular person at the keyboard. Usage analysis may provide other identity clues, but that, too, isn't foolproof. Establish the limits to what an examiner can say with certainty, and afford the examiner an opportunity to concede those limits or overreach them.

Missing in Action

When hard drives were smaller, it was possible to thoroughly examine them by looking through the data. It was a tedious process, to be sure, and one where it was easy to grow tired and overlook something. Still, it was a pretty reliable process. Hard drives have grown to gargantuan volumes, e.g., the 500-gigabyte hard drive in my current system is *25,000 times larger* than the 20-megabyte drive in my first personal computer. It's all but impossible in the usual engagement for an examiner to look at all the data on the drive. It's overwhelming to thoroughly examine just the places where data most often hides.

Consequently, examiners must rely upon software tools to get the job done. Keyword searches are an integral part of computer forensic examinations and entail an examiner entering key words, phrases or word fragments into a program which then scours the drive data to find them. False positives or negatives are an issue, along with the literal way computers approach searches. A human eye will see the word "Confidential" though it be written C.o.n.f.i.d.e.n.t.i.a.l, Confidentail or _onfidential, but a computer can't make the connection unless it's been programmed to identify common variants or uses more advanced search algorithms. When the matter in dispute hinges on what *wasn't* found on the drive, the ingenuity and diligence applied to the search may be fodder for cross-examination. Of

course, whatever points you score forcing the examiner to admit he didn't pursue certain searches can be lost when the witness returns the next day having completed those searches without finding anything.

Dealing with Digests

Disk drives are so vast and operating systems so complex, how can a forensic examiner be certain that someone hasn't slipped in incriminating data? A forensic examiner might respond that, when acquired, the data on the hard drive is "hashed" using sophisticated encryption algorithms and a message digest is calculated, functioning as a fingerprint of the drive. Once hashed, the chance that tampering would not be detected is one in 340 undecillion--and that's one in 340 followed by *36 zeroes*! That's FAR more reliable than DNA evidence! It's an impressive assertion, and even true…to a point. Hash collisions have been engineered in recent years for the common MD5 hash algorithm. Though these proven defects in the algorithm don't seriously imperil its value in forensics, an examiner who testifies hash collisions are "impossible" is one not keeping abreast of the discipline.

Contrived collisions aside, drive hashing and the creation of those message digest "fingerprints" is indeed one of the slickest tools in a forensic examiner's arsenal. The reliability assertion is genuine (though the probabilities vary among commentators). But, the probative value of hashing depends upon the points in time during the acquisition and analysis process when hashing is done and, ultimately, upon the veracity of the examiner who claims to have hashed the drive. Two identical message digests of a drive tell you only that no tampering occurred between the time those two digests were computed, but tell you nothing about tampering at other times. If a drive is altered, then hashed, subsequent hashes can be a perfect match without revealing the earlier alteration. Likewise, an earlier hash can't tell you anything about subsequent handling; at least, not until the drive is hashed again and the digests compared. The point is, be sure you know when the hashing was done and where that activity falls with respect to the entire chain of custody. Also, consider whether the hashing was done by someone telling the truth. A cross-examiner might score some cheap points by getting the witness to attest to the importance of hashing, and then asking the witness to explain the mathematical process by which such a critical step is accomplished. Some experts understand cryptography and can explain it, but I suspect their ranks are small.

Pornographic Images

Aside from the scourge of child pornography, the law makes a person's proclivity for pornography their own affair; unless, of course, that person is my employee and dumps their trash on my computer system. Pornography, the bread-and-butter of law enforcement computer forensic examinations, is a civil litigation issue in cases of wrongful termination or harassment. When used as grounds for discipline or termination, or when the presence of smut will otherwise be used to impeach, it's essential to be able to reliably link the objectionable imagery to its true owner.

It's a simple matter to load a computer with dirty pictures unbeknownst to the owner. One reprehensible way to do this is to embed the pictures in an innocuous e-mail but specify the dimensions of the image to be a single pixel. That way, all of the image data gets

downloaded to their computer, but the recipient doesn't see a thing. The porn file or other electronic contraband now resides on the recipient's computer and there's no reason to believe the recipient didn't put it there unless you go looking for other avenues. Though rarer in fact than in contention, the same insidious result is achieved using an outwardly benign web site or precipitated by a malevolent virus. The upshot is that an amateur examination of the computer reveals megabytes of porn or other incriminating material, and management goes ballistic.

Fortunately, a skilled and cautious investigator can spot the difference between an unwitting victim and avid accumulator. Sheer volume is a factor, but the location of the images and efforts to conceal or delete them, as well as their creation and access times, format and context all tend to reveal the truth. Any skilled examiner should be able to authoritatively address the question, "How do you know my client put these files on the computer?" A reply of, "It was his computer and the pictures were on it" is always an inadequate explanation.

Checklists and Notes

Thoroughly analyzing a hard drive is a long, detailed and complicated process. It's easy to overlook or fail to follow up on something. Those who undertake other critical, complex and repetitive tasks are aided by checklists (survival tip: never fly with a pilot who doesn't take the preflight checklist very seriously). However, computer forensic analysts are sometimes loathe to use checklists for fear criminal defense lawyers will crucify the examiner for skipping a step, even when the shortcut is justified. Spanning the realms of art and science, and dealing as we do with human frailty, computer forensics examiners are aided by instinct and gut feeling--skills which don't lend themselves to checklists.

The twin goals of cross-examination are to secure helpful concessions and blunt the impact of whatever hurts your case. If an examiner uses checklists or a published methodology, obtain copies of those items and search for the overlooked step suggesting carelessness. If the examiner doesn't use some written system to insure a consistent analytic approach, then the examiner might be taken to task for that. An experienced witness isn't going down in flames this way, but it may flush out charlatans and novices.

In a similar vein, all the literature emphasizes, and veteran examiners agree upon, the importance of carefully documenting a forensic analysis. If the witness claims to have no notes (including electronic logs), there's something amiss. Inquire if the witness' analysis tools track activities like keyword searches and whether those logs have been saved or altered. Obtain and check logs for matters overlooked, such as results omitted from reports or incomplete follow up.

Get Help

Cross-examining a technical expert on matters you don't understand is playing with fire. Though you can't quickly become the equal of someone who's spent years mastering an esoteric specialty, you can learn a great deal about one or two specific aspects of that specialty. Pick your battles and do your homework to win the day. You can pick up the fundamentals from my articles at www.craigball.com. For top notch online information about computer forensics, visit the Electronic Evidence Information Center at www.e-

evidence.info/index.html or the resource library areas of the following vendor sites: New Technologies, Inc. (www.forensics-intl.com), Computer Forensics, Inc. (www.forensics.com), Guidance Software (www.guidancesoftware.com) or AccessData (www.accessdata.com).

Finally, don't charge into battle alone. Even if you haven't invested in your own computer forensic analysis, it might be worthwhile to engage an expert to review the other side's findings or back you up at deposition or trial.

CRAIG BALL

Trial Lawyer & Technologist
Computer Forensic Examiner
Author and Educator

1101 Ridgecrest Drive
Austin, Texas 77486
E-mail: craig@ball.net
Web: craigball.com
Office: 512-514-0182
Fax: 512-532-6511
Mobile: 713-320-6066

Craig Ball is a Board Certified trial lawyer, certified computer forensic examiner and electronic evidence expert He has dedicated his career to teaching the bench and bar about forensic technology and trial tactics. After decades trying lawsuits, Craig now limits his practice solely to serving as a court-appointed special master and consultant in computer forensics and electronic discovery, and to publishing and lecturing on computer forensics, emerging technologies, digital persuasion and electronic discovery. Craig's award-winning e-discovery column, "Ball in Your Court," appears in Law Technology News. He has consulted or served as a testifying expert in computer forensics and electronic discovery in some of the most challenging and well-known cases in the U.S. Named as one of the Best Lawyers in America and a Texas Superlawyer, Craig is a recipient of the Presidents' Award, the State Bar of Texas' most esteemed recognition of service to the profession and of the Bar's Lifetime Achievement Award in Law and Technology.

EDUCATION

Rice University (B.A., triple major, English, Managerial Studies, Political Science, 1979); University of Texas (J.D., with honors, 1982); Oregon State University (Computer Forensics certification, 2003); EnCase Intermediate Reporting and Analysis Course (Guidance Software 2004); WinHex Forensics Certification Course (X-Ways Software Technology AG 2005); numerous other classes on computer forensics and electronic discovery.

SELECTED PROFESSIONAL ACTIVITIES

Law Offices of Craig D. Ball, P.C.; Licensed in Texas since 1982.
Board Certified in Personal Injury Trial Law by the Texas Board of Legal Specialization
Certified Computer Forensic Examiner, Oregon State University and NTI
Certified Computer Examiner (CCE), International Society of Forensic Computer Examiners
Admitted to practice U.S. Court of Appeals, Fifth Circuit; U.S.D.C., Southern, Northern and Western Districts of Texas.
Member, Editorial Advisory Board, Law Technology News and Law.com (American Lawyer Media)
Board Member, Georgetown University Law School Advanced E-Discovery Institute
Member, Sedona Conference WG1 on Electronic Document Retention and Production
Special Master, Electronic Discovery, Federal and Harris County (Texas) District Courts
Instructor in Computer Forensics, United States Department of Justice
Instructor, Cybercrime Summit, 2006, 2007
Special Prosecutor, Texas Commission for Lawyer Discipline, 1995-96
Council Member, Computer and Technology Section of the State Bar of Texas, 2003-
Chairman: Technology Advisory Committee, State Bar of Texas, 2000-02
President, Houston Trial Lawyers Association (2000-01); President, Houston Trial Lawyers Foundation (2001-02)
Director, Texas Trial Lawyers Association (1995-2003); Chairman, Technology Task Force (1995-97)
Member, High Technology Crime Investigation Association and International Information Systems Forensics Assn.
Member, Texas State Bar College
Member, Continuing Legal Education Comm., 2000-04, Civil Pattern Jury Charge Comm., 1983-94,State Bar of Texas
Life Fellow, Texas and Houston Bar Foundations
CLE Course Director: E-Discovery A-to-Z (NY, Chicago, SF, Boston, Washington, D.C., Minneapolis, Miami, Houston, Seattle, Atlanta, Denver, Philadelphia) 2004-7; Electronic Evidence and Digital Discovery Institute 2004-7; Advanced Evidence and Discovery Course 2003; 2002; Enron—The Legal Issues, 2002; Internet and Computers for Lawyers, 2001-02; Advanced Personal Injury Law Course, 1999, 2000; Preparing, Trying and Settling Auto Collision Cases, 1998.

Member, SBOT President's "Vision Council" on Technology, 1999-2000; Strategic Planning Committee Liaison, 2001-02; Corporate Counsel Task Force 2001-02

ACADEMIC APPOINTMENTS AND HONORS

2006 Recipient of the State Bar of Texas CTS Lifetime Achievement Award for Law and Technology
The March 2002 CLE program planned by Mr. Ball and Richard Orsinger entitled, "Enron—The Legal Issues" received the Best CLE of 2002 award from the Association for Legal Education
National Planning Committee, Legal Works 2004 (San Francisco)
Recipient, State Bar of Texas Presidents' Award (bar's highest honor), 2001
Faculty, Texas College of Trial Advocacy, 1992 and 1993
Adjunct Professor, South Texas College of Law, 1983-88
Listed in "Best Lawyers in America" and Selected as a "Texas Super Lawyer," 2003-2006
Rated AV by Martindale-Hubbell

LAW RELATED PUBLICATIONS AND PRESENTATIONS

Craig Ball is a prolific contributor to continuing legal and professional education programs throughout the United States, having delivered over 450 presentations and papers. Craig's articles on forensic technology and electronic discovery frequently appear in the national media, including in American Bar Association, ATLA and American Lawyer Media print and online publications. He also writes a monthly column on computer forensics and e-discovery for Law Technology News called "Ball in your Court," which received which is the 2007 Gold Medal honoree as "Best Regular Column" as awarded by Trade Association Business Publications International. It's also the 2007 Silver Medalist honoree of the American Society of Business Publication Editors as "Best Contributed Column" and their 2006 Silver Medalist honoree as "Best Feature Series" and "Best Contributed Column." The presentation, "PowerPersuasion: Craig Ball on PowerPoint," is consistently among the top rated continuing legal educational program from coast-to-coast.