CHAPTER 2

Key Technical Concepts

Information in This Chapter:

- Basic Computer Operation
- Bits & Bytes
- File Extensions and File Signatures
- How Computers Store Data
- Random Access Memory
- Volatility of Data
- The Difference Between Computer Environments
- Active, Latent, and Archival Data
- The Difference Between Allocated and Unallocated Space
- Computer File Systems

INTRODUCTION

Intimate knowledge of the inner workings of a computer is critical for the digital forensics practitioner. It's this knowledge that permits us to conduct a thorough examination of the evidence and render an accurate opinion. Simply put, we can't do our job without it. Not all processes and hardware hold the same value forensically. Memory and storage play a major role in almost any examination. The processor or CPU, by contrast, plays little if any role. This chapter takes a broad look at some of the technical details of basic computing. Its focus will be on the major areas that impact an investigation. There is no substitute for the mastery of this material. Our responsibilities as an expert witness include explaining technical subject matter in a way that the average person is able to understand.

BITS, BYTES, AND NUMBERING SCHEMES

To the computer, things are pretty black and white. It's all about the 1s and 0s. Computers use a language called **binary**. In binary, there are only two possible outcomes: a 1 or a 0. Each 1 or 0 is called a bit. In mathematical terms, binary is classified as a base 2 numbering system. In comparison, we use a base 10 numeral system known as **decimal**. Decimal uses numerals 0–9. To speed things up,

The Basics of Digital Forensics. DOI: 10.1016/B978-1-59749-661-2.00002-4

computers work with larger collections of bits. These larger chunks of data are called **bytes**. A byte is made up of eight bits. It looks like this: 01101001.

How do bytes relate to letters and numbers? Each letter, number, space, and special character is represented by a single byte. For example, using the ASCII character set 01000001 represents an uppercase "A," while a lowercase "a" is 01100001.

Let's do a little experiment so that you can see this in action. Open a new text document (using a plain text editor, not a word processing application like MS Word) on your computer and type the phrase "Marshall University Digital Forensics." Now, count all the letters and spaces. Next, save and close the new text file to your desktop. Right click on the file and select properties. What's the file size? It should be 26 bytes, which is also the exact number of letters and spaces.

To get a broader perspective, let's look at all of the binary necessary to represent our sample phrase "Marshall University Digital Forensics":

```
01001101011000010111001001110011011010000110000101101100011011000010000000101010101110111001101001011101100111001011100100111001101101001011101000111100100100000010001000110100101100111011010010111010000110000101101100001000000100011001101111011100100110010101101110011010010110000110100111001101101001011000011011100110
```

At first glance, that's a little tough to read, no doubt. Fortunately, there is a shorthand that we can use to make this more readable. This shorthand is called **hexadecimal**.

Hexadecimal

Hexadecimal, or hex, is a base 16 system that is an expedient way to express binary numbers. Hex is expressed using the numerals 0–9 and the letters A–F. An uppercase "M" is expressed as 4D in hexadecimal. A lowercase "a" is 61. Quite often you will see a hexadecimal number expressed with the prefix 0x. This prefix or the suffix "h" is used to designate or identify it as a hexadecimal or base 16 number. Here is the same phrase (Marshall University Digital Forensics) expressed in hexadecimal:

```
4d 61 72 73 68 61 6D 6C 20 55 6E 69 76 65 72 73 69 74 79
20 44 69 67 69 74 61 6D 20 46 6F 72 65 6E 73 69 63 73
```

If you look closer, you'll see the number "20" repeated throughout the string. The number "20" in hex represents a space.

Binary to Text: ASCII and Unicode

So how do these 1s and 0s end up as As and Bs? Computers use encoding schemes to convert binary into something humans can read. There are two

encoding schemes we need to be concerned with, **ASCII** and **Unicode**. ASCII, the American Standard Code for Information Interchange, is the encoding scheme used for the English language. ASCII defines 128 characters, of which only 94 are actually printable. The rest are control characters used for spacing and processing. In contrast, **Unicode** is intended to represent all of the world's languages and consists of thousands of characters (Unicode Inc., 2010).

So, how is this relevant to digital forensics? In many instances, examiners must look at the data at the "bit" and "byte" level to find, extract, and interpret the evidence. This is most evident in a process called **file carving**. File carving is done to locate and mine out files from amorphous blobs of data, like the unallocated space (also known as drive-free space). The first step in the file carving process is to identify the potential file. Normally, the file is identified by the header, if it has one. Once the footer is found, the file can be extracted through a simple copy and paste as long as it is continuous. A fragmented file is far more difficult to recover (Casey, 2011). Having the ability to interpret binary and hex makes file carving possible.

FILE EXTENSIONS AND FILE SIGNATURES

Fundamentally, **files** are strings or sequences of bits and bytes. Identifying a file can be done in a couple of different ways. **File extensions** are the most common. As users, we usually identify the file type by the file extension, if the system is configured. An operating system can be set such that file extensions are hidden. File extensions are the suffixes added to the end of a computer file name, indicating its format. Examples would include .docx and .pptx (for the latest versions of Microsoft Word and PowerPoint, respectively).

For our purposes, a file extension isn't the most reliable way to identify it. The file extension is very easily changed, requiring only a mouse click and a couple of keystrokes. You can try this yourself. In Windows, simply right click on the file name and rename it, changing the extension. Let's say we change the extension of a Word file to that of an image, JPEG for example. This is easily accomplished. On a Windows machine, simply click, slight pause, click again. On a Mac, it's click + Return. What happens when we try to open that file? Nothing. It won't open. Change it back and it opens right up.

Some people will attempt to take advantage of this ability to change file extensions as a way to conceal data, hiding them in plain sight. Forensically, this approach is not very effective. Forensic tools identify files based on the header, not the file extension. Many tools will even separate out those files whose header does not match the extension, making them easily discovered. This comparison is generally known as **file signature analysis**. Figures 2.1 and 2.2 illustrate what happens when a file extension is changed.

FIGURE 2.1
Here we've changed the file extension on "Smoking Gun.docx" to .mp3. Note that the icon has changed. Graphic courtesy of Jonathan Sisson.

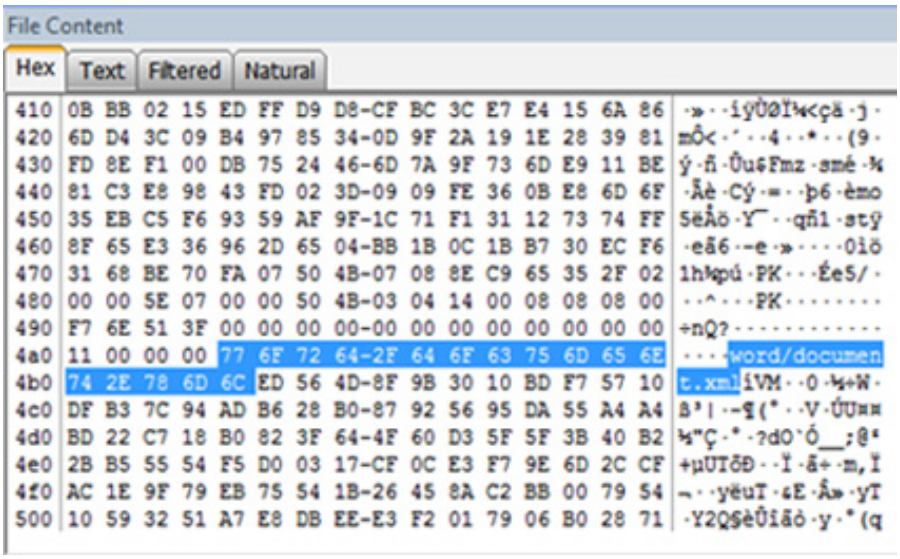

FIGURE 2.2
Here is the hexadecimal view of "Smoking Gun.mp3." Note the highlighted file header showing this is actually a Word document. Graphic courtesy of Jonathan Sisson.

STORAGE AND MEMORY

Where and how data are stored and written is one of the major fundamental concepts that must be learned. There is more that one way to write data. Today, data are generally created in three different ways: **electromagnetism**, **microscopic electrical transistors (flash)**, and **reflecting light** (CDs, DVDs, etc). Storage locations inside a computer serve different purposes. Some are for the short term, used to temporarily hold the data that the computer is using at the moment. The other is for more permanent, long-term keeping.

Magnetic Disks

Most drives in today's computers read and write data magnetically. They will render each particle either magnetized or not magnetized. If the particle is magnetized, it's read as a 1. If not, it's read as a 0. The drives themselves are usually made up of aluminum platters coated with a magnetic material. These platters spin at very high speeds. The platters spin in the neighborhood of 7,000 rpm to 15,000 rpm. The speed could even be greater for high-end drives. These heavy-duty drives are typically found in servers or professional grade workstations. From a forensic standpoint, faster drive speeds can result in faster acquisitions.

Let's look at the major parts of a standard hard drive. The platters revolve around a small rod called a spindle. The data are physically written to the platter using a read/write head attached to an actuator arm, which is powered by the actuator itself. The actuator arm moves the head across the platter(s), reading and writing data. The read/write head floats on a cushion of air. The read/write head, as it's called, is barely floating above the platter surface, at a height less than the diameter of a human hair. These devices are really pretty amazing. Figure 2.3 shows

FIGURE 2.3
The inside of a typical magnetic drive.

us the inside of a typical magnetic drive. We can clearly see the platters, actuator arm, and the read/write head.

Flash Memory

Flash memory is used in a wide range of devices. Thumb drives and memory cards provide reliable storage in a very portable package, allowing us to take more pictures and take our files on the road. Unlike other kinds of memory, flash memory retains our data even without electricity. Flash is made up of **transistors**. Each transistor is either carrying an electric charge or it isn't. When the transistor is charged, it is read as a "1"; without a charge it's read as a "0."

Flash based hard drives are starting to become more and more common. Unlike magnetic drives, flash drives are solid state, meaning that they have no moving parts. They are often referred to as an SSD or **"Solid State Drive."** They offer several significant advantages including increased speed, less susceptibility to shock, and lower power consumption.

SSDs will play a major role in computing and digital forensics going forward. Although these devices offer improved performance, they also present a major challenge to digital forensics. We'll take a deeper look at the momentous challenge presented by SSDs in Chapter 11.

Optical Storage

Optical media read and write data using a laser light along with a reflective material incorporated into optical discs. Optical discs are made of a polycarbonate base covered by a thin layer of aluminum. The disc is then coated with a clear acrylic material for protective purposes. During the manufacturing process, the disc's surface is embossed with tiny bumps. This series of bumps form one long, single, spiral track. A laser projects a highly focused beam of light onto the track. The light is reflected differently from the bumps and the spaces in between, called "**lands**." This change in reflectivity is what the system reads as binary (Brain). The most common types of optical storage media include CDs, DVDs, and Blu-ray discs (Brain).

Volatile versus Nonvolatile Memory

Memory and **storage** are two terms that are somewhat synonymous when it comes to computers. They both refer to internal places where data are kept. Memory is used for the short-term storage, while storage is more permanent. No matter what you call it, there is a significant difference between the two, especially from a forensic perspective. That difference lies in the data's volatility. Data in RAM exist only as long as power is supplied. Once the power is removed (i.e., the machine is turned off), the data start to disappear. This behavior makes this kind of memory volatile. In contrast, files saved on your hard drive remain even after the computer is powered down, making it nonvolatile (Cooper, 2004).

RAM stores all the data that are currently being worked on by the Central Processing Unit (CPU). Data are fed from the RAM to the CPU, where they are executed. Traditionally, forensic analysis of a computer focused on the hard drive, as much of the evidence can be found there. Today, we're finding that's not always the case. Some instant messaging applications, for example, don't write to the hard drive unless the logging feature is turned on. AOL Instant Messenger and MSN fall into that category. So, if logging is off (which it is by default), the only evidence will be found in RAM while the machine is running.

COMPUTING ENVIRONMENTS

Not all **computing "environments"** are created equal. There are substantial differences between them. We can encounter individual computers, networks of various sizes, or even more complex systems. These disparities will have a significant impact on your collection process, where you look for data, the tools you will use, and the level of complexity required. An accurate clarification of the environment is useful to have right from the start of an investigation, even before you respond to a scene. Environments can be broken down into four categories: stand-alone, networked, mainframe, and the cloud.

A stand-alone computer is one that is not connected to another computer. These are the easiest to deal with and investigate. Possible locations for evidence are reasonably confined. Stand-alone systems are routinely encountered in residences such as apartments and houses.

A networked computer is connected to at least one other computer and potentially many, many others. This escalates the complexity as well as the places evidence could be found. We now can see files and artifacts normally found on the local machine spread out to servers or other machines. This environment introduces a variety of variables into the equation. Even though networks are more commonly found in a business setting, they are found more and more in homes.

Unlike a stand-alone machine, a **mainframe system** centralizes all of the computing power into one location. Processors, storage, and applications can all be located and controlled from a single location.

Cloud Computing

You may not be familiar with the term "**cloud computing,**" but if you use Gmail, Facebook, or Twitter, you're already using it. Cloud computing is a hot topic these days, garnering much attention from both the IT and business communities. This "new" model of computing is very similar in many respects to the mainframe systems of old. Like the mainframe, the computing resources are moved from the local machine to some other centralized place.

The cloud model presents some very interesting features that make it attractive to businesses, especially from a cost perspective. The cloud offers software along with computing infrastructure and platforms on an elastic, pay-per-use model. This affords companies the luxury of only paying for what they use.

Technology behemoths such as Microsoft, Google, and Amazon are just three of the companies that are jumping on the bandwagon offering cloud services. Cloud services include **Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).** All of these are delivered over the Internet. In the cloud, customers only pay for the resources they actually use, just like the way we pay for our water and electricity.

IaaS

With IaaS, organizations outsource their hardware needs to a service provider. This would include everyday hardware needs such as servers, storage, and the like. The associated costs for running and maintaining the hardware are paid by the provider.

PaaS

Programmers develop their software to function in specific computing environments (operating system, services, etc.). PaaS gives developers the ability to rent the environment (hardware, operating systems, storage, servers, etc.) on an "as-needed" basis. PaaS provides excellent flexibility in that the operating system can be modified or upgraded frequently.

SaaS

In the cloud, SaaS provides applications on demand to customers over the Internet. These applications are hosted and maintained by the service provider.

The cloud represents a huge challenge to the digital forensic community, from both a technical and a legal standpoint. Technically, the cloud presents a very complicated, virtualized environment that frustrates if not downright negates many routine forensic procedures. Legally, it can be a jurisdictional nightmare. In the cloud, data know no bounds. The evidence can literally be in the next state or a foreign country halfway around the globe. We'll look closer at the cloud and its impact on forensics in Chapter 11.

DATA TYPES

Data can be lumped into three broad categories: active, latent, and archival. Looking at data in this way helps in clarifying their location, how they're accounted for by the file system, how they can be accessed by the user, and so on. It also helps to narrow down the cost and effort required to recover the data in question.

Active Data

Active data are the data that we use every day on our computers. The operating system "sees" and tracks these files. You can locate these files using Windows Explorer. These are the files that reside in the allocated space of the drive. These data can be acquired with standard forensic cloning techniques.

Latent Data

Data that has been deleted or partially overwritten are classified as **latent**. These files are no longer tracked by the operating system and are therefore "invisible" to the average user. Go looking for one of these files with Windows Explorer and you won't find it. A bit stream or forensic image is required to collect these data.

Archival Data

Archival data, or backups, can take many forms. External hard drives, DVDs, and backup tapes are just a few examples. Acquisition of archival data can range from simple to extremely complex. The type and age of the backup media are major factors in determining the complexity of the process.

Backup tapes can present some very big challenges, especially if they were made with software or hardware that is no longer in production. Tapes are created using specific pieces of hardware and software. These same tools will be needed to restore the data into a form that can be understood and manipulated. Where it gets really exciting is when the hardware and software are no longer in production. It could be an older version of the software is no longer available or the company is no longer in business. This is known as **legacy data**. What do you do if you no longer have and can't get access to the necessary tools to restore the data? Sometimes eBay can save the day.

FILE SYSTEMS

With all the millions or billions of files floating around inside our computers, there has to be some way to keep things neat and tidy. This indispensible function is the responsibility of the **file system**. The file system tracks the drive's free space as well as the location of each file. The free space, also known as unallocated space, is either empty or the file that previously occupied that location has been deleted.

There are many different types of file systems. Some of the most commonly encountered by forensic examiners include FAT, NTFS, and HFS+. Let's take a closer look:

File Allocation Table (FAT) is the oldest of the common files system. It comes in four flavors: FAT12, FAT16, FAT32, and FATX. Although not used in the latest operating systems, it can often be found in flash media and the like.

The New Technology File System (NTFS) is the system used currently by Windows 7, Vista, XP, and Windows Server. It's much more powerful than FAT and capable of performing many more functions. For example, "NTFS can automatically recover some disk-related errors, which FAT32 cannot," it provides better support for larger hard drives, and better security through permissions and encryption (Microsoft Corporation).

Hierarchical File System (HFS+) and its relatives HFS and HFSX are used in Apple products. HFS+ is the upgraded successor to HFS. This newer version offers several improvements including improved use of disk space, cross-platform compatibility, and international-friendly file names (Apple, Inc., 2004).

ALLOCATED AND UNALLOCATED SPACE

Before we get much further, it's time we talk about how the computer views the space on a hard drive. Generally speaking, the file system categorizes all of the space on the hard drive in one of two ways. The space is either **allocated** or **unallocated** (there are a few exceptions; see the side bar on Host Protected Areas). Put another way, either the space is being used or it's not. Windows can't see data in this unallocated space. To the Operating System, files located in unallocated space are essentially invisible. It's important, however, to understand that "not used" does not always mean "empty."

MORE ADVANCED

Host Protected Area (HPA) and Device Configuration Overlays (DCO)

Host Protected Areas (HPAs) and **Device Configuration Overlays** (DCOs) refer to hidden areas on a hard drive that are often difficult to detect. These areas are created by manufacturers that can be "accessed, modified, and written to by end users using specific open source and freely available tools, allowing data to be stored and/or hidden in these areas" (Gupta, Hoeschele, & Rogers, 2006). HPAs can contain diagnostic tools, an operating system for recovery purposes, and so on. It's rare that the HPA is used by suspects to conceal data.

Data Persistence

Like a telemarketer, data on a hard drive are pretty persistent. It's not as easy to get rid of as you may think. Deleted files will sit there until they're overwritten with more data. You might be asking yourself, "So how long does that take?" The answer is, it depends (which, by the way, is one of the most popular answers in digital forensics). With the massive amount of storage space available on today's hard drives, a file stands a good chance of never being overwritten. Your bachelor (or bachelorette) party pictures could remain on your hard drive for a long, long time. Just keep that in mind before you run for public office.

Remember, the file system's job is to keep track of all files and storage space. The file system keeps things nice and orderly. Think of a file system as an index in the back of a book. When looking up a particular subject, we flip through the index until we find the term we're looking for. Our handy index then gives us the page number and off we go. The file system works basically the same way. Using the book analogy again, deleting a file would be akin to removing the entry from the book's index. Although our subject is no longer referenced in the index, the page and all its content are still in the book, intact and untouched.

You may be surprised to know that when you save your file, it's not necessarily stored in one place. In fact, your spreadsheet could be scattered all over the platter(s) of your hard drive. Strange, huh? You would think as orderly as computers are, that wouldn't be the case.

The file system's job is to keep track of these separate clusters so they can be reassembled the next time you open that file. Have you ever "defragged" your hard drive? If you have, you were simply moving these disparate pieces as close together as possible. Moving them closer together speeds things up for your computer. The closer they are, the faster they can be put together and made available to you. Some crooked individuals may attempt to destroy data using the defragging process. In Chapter 6, we'll see how that may or may not be effective.

Files that are overwritten are generally considered to be unrecoverable. But all is not lost (pardon the pun). Like many rules in life, there are exceptions and this is one of those. It is possible that the new file assigned to that space won't need all of it. If that's the case, the original file is only *partially overwritten*. The piece that remains *can* be recovered and could contain information we can use. This remaining space is called **slack space**. Before we take a little closer look at slack space, we're going to have to get a little more technical. So, get your "nerd on" and follow along.

HOW MAGNETIC HARD DRIVES STORE DATA

We need to understand how the computer stores your files. Computers store your data in defined spaces called **sectors**. Think of sectors as the smallest container a computer can use to store information. Each sector holds up to 512 bytes of data as illustrated in Figure 2.4. It can hold less, but it can't hold more.

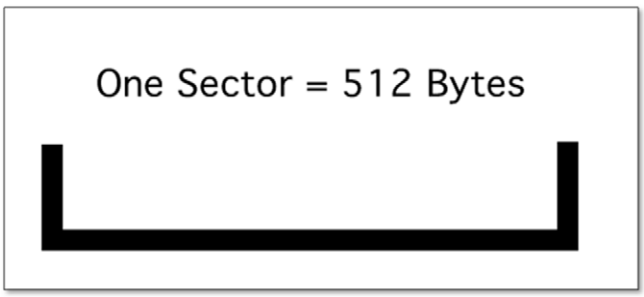

FIGURE 2.4
One sector.

While a sector is the smallest container of data, a computer's operating system only stores data as clusters. Clusters are comprised of multiple sectors. In this example our clusters contain frour sectors. Each sector can hold up to 512 bytes of data, giving the clusters the storage capacity of 2048 bytes. See figure 2.5.

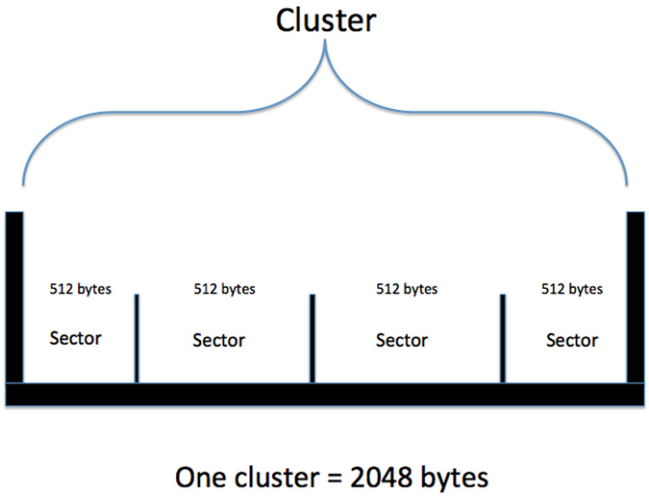

FIGURE 2.5
A sample cluster containing four 512 byte sectors, giving it a maximum capacity of 2048 bytes

It's important to remember that computers write data to the drive in clusters. If the file is larger than a single cluster, the system assigns it an additional cluster even though a portion of that cluster may not be used. Let's work through a little hypothetical exercise to better illustrate this concept.

Suppose we save our master criminal plan to our hard drive. We'll call it "evidence1.doc". It just so happens to be 2304 bytes in size. Since it's larger than our cluster size limit (2048 bytes) it's assigned to two separate, unallocated clusters (in this example, clusters 5245 and 5246). You'll also notice that our file only uses a portion of the first sector in the second cluster. Since the machine has to write 512 bytes at a time, it fills that leftover space with zeros. See figure 2.6.

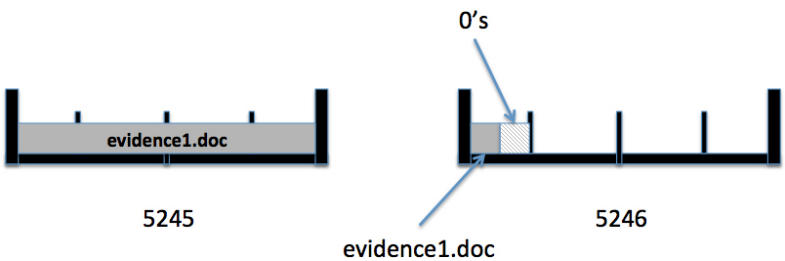

FIGURE 2.6
The file, evidence1. doc is saved to the hard drive. It's assigned to clusters 5245 and 5246. Note that the rest of cluster 5246 is left unallocated.

What about the last three sectors in cluster 5246 that weren't used? The answer is nothing. As we'll see in just a bit, this system behavior can leave some evidence behind. See figure 2.7

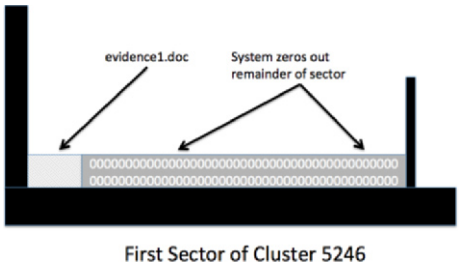

FIGURE 2.7
The unused portion of the last sector occupied by "evidence1.doc" is filled with zeros because the computer only writes data 512 bytes at a time.

After watching Abby and McGee work their magic on NCIS, we start to have second thoughts. We decide it's probably better not to have that file on our computer. So we hit the delete key, sending the "evidence1.doc" to the recycle bin. With a sly grin we empty the recycle bin, content in the belief that "evidence1.doc" is now residing in digital oblivion. But wait, not so fast. The problem for us as bad guys is that unbeknownst to us, our incriminating file is STILL on the drive. It will remain in those two clusters until it's been overwritten by another file. Given the size of today's dives, that could take a very, very long time. Using standard forensic tools, we can recover any part of the document that hasn't been overwritten. Figure 2.8 depicts our two clusters after the recycle bin has been emptied.

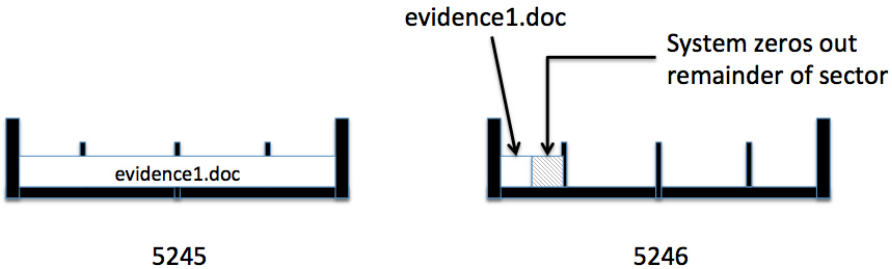

FIGURE 2.8
The file evidence1.doc has been deleted. Clusters 5245 and 5246 are now marked as unallocated (available). Notice that even though evidence1.doc has been deleted, it's still on the hard drive.

Now for some really cool forensic stuff. Even if the clusters containing our evidence are allocated to another file, all is not lost. It's still possible that we can extract a portion of the original file. Here's how it works. Two days later, we save another file to our drive. We'll call this one "evidence2.doc". It's only 768 bytes in size so it only takes one cluster to hold it. The system sees that cluster 5245 is available and decides to put it there. Remember, evidence1.doc is still sitting in the cluster even though it's been "deleted". The system writes "evidence2.doc" to the first sector and part of the second. It then does its normal thing and fills the remainder of that second sector with zeros. So what happens to the rest of evidence1.doc? When we first saved it, it took up all of cluster 5245. Our new file (evidence2.doc) has overwritten only PART of evidence1.doc. The remnants of evidence1.doc that sits in the last two sectors can be recovered! See figure 2.9.

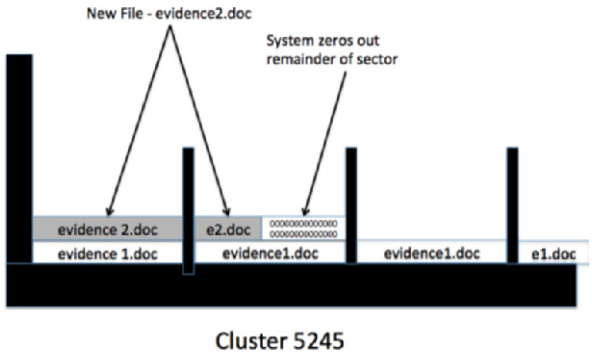

FIGURE 2.9
evidence2.doc is saved over evidence1.doc, overwriting the much of the original file.

To recap, the only the first 780 bytes of our original file have been overwritten. Some quick math tells us that there are still 244 bytes of our original file remaining. Those 244 remaining bytes comprise what's known as slack space. The slack space, depicted in figure 2.10, is the difference between the space that is assigned and the space that is actually used.

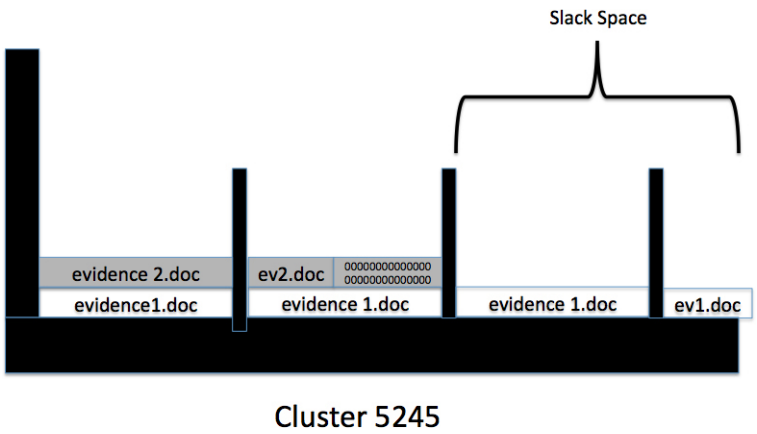

FIGURE 2.10
Note the new file, "evidence2.doc" only overwrites a portion of "evidence1.doc." The data in the remaining two sectors are still intact. This fragment of data can be recovered and could contain useful evidence."

So, out of the slack space we can recover fragments of the previous file. It may not be useful. But then again, it just might. It could be part of an incriminating spreadsheet, email or picture. These fragments could contain just enough of an email to identify the sender or the senders IP address. A partial picture of the victim could link them to the suspect. Slack space can't be accessed by the user or the operating system. As such, this evidence exists unbeknownst to all but the most tech-savvy suspects.

Unfortunately, recovering evidence from slack space may very well become a thing of the past. We'll explore that bad news more in Chapter 11, "Looking Ahead: Challenges and Concerns."

SUMMARY

In Chapter 2 we took a closer look at how computers store data in different forms including magnetic, optical, flash, and others. Each of these storage methods is different and those differences have forensic implications. Computers operate with both memory and storage. While they sound similar, their intended purposes are distinctly different. Memory holds the data that the computer is actively working on at the moment. It's volatile, meaning that it holds data as long as it has power. When power is removed, the data begins to go away. The RAM in your computer is used for memory.

In contrast, storage is used for the long-term storing of data. Storage is considered non-volatile because the data remains even if the device loses power. Your hard drive is an example of storage.

A computer's file system is at the heart of how it saves and retrieves data. File systems keep track of the various pieces of data that must be found and reconstituted in order to open a file. There are multiple file systems in use today, each with their own way doing things.

Not all computing environments are the same. Some are relatively simple, others much more complex. Stand-Alone computers, networks, and the cloud were covered in this chapter.

As forensic examiners, we must have command of this material so that we can explain it to the average person. It is these "average people" that make up our juries.

References

Apple, Inc. (2004, March 5). *Technical Note TN1150 HFS Plus Volume Format.* Retrieved August 10, 2011, from: http://developer.apple.com/library/mac/#technotes/tn/tn1150.html

Casey, E. (2011). *Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet*. Waltham, MA: Academic Press.

Casey, E. (2009). *Handbook of Digital Forensics and Investigation*. Burlington, MA: Academic Press.

Cooper, B. (2004, August). *What Is the Difference Between Memory and Storage?* Retrieved August 10, 2011, from: http://searchstorage.techtarget.com/answer/What-is-the-difference-between-memory-and-storage

Dale, N. (2009). *Computer Science Illuminated, Fourth Edition*. Sudbury, MA: Jones and Bartlett.

Gupta, M. R., Hoeschele, M. D., & Rogers, M. K. (2006). Hidden Disk Areas: HPA and DCO. *International Journal of Digital Evidence, 5* (1).

Microsoft Corporation. (n.d.). *Comparing NTFS and FAT File Systems*. Retrieved August 10, 2011, from: http://windows.microsoft.com/en-US/windows-vista/Comparing-NTFS-and-FAT-file-systems

SearchStorage.com. (2000, December). *Optical Media*. Retrieved August 10, 2011, from: http://searchstorage.techtarget.com/definition/optical-media

Unicode Inc. (2010, September 17). *What Is Unicode?* Retrieved August 10, 2011, from: http://www.unicode.org/standard/WhatIsUnicode.html