chapter 17

Computer Forensics

By Andrew W. Donofrio

Key Terms

bit

byte

central processing unit (CPU)

cluster

file slack

hard disk drive (HDD)

hardware

latent data

Message Digest 5 (MD5)/secure hash algorithm (SHA)

motherboard

operating system (OS)

partition

RAM slack

random-access memory (RAM)

sector

software

swap file

temporary files

unallocated space

visible data

Learning Objectives

After studying this chapter you should be able to:

- List and describe the hardware and software components of a computer
- Understand the difference between read-only memory and random-access memory
- Describe how a hard disk drive is partitioned
- Describe the proper procedure for preserving computer evidence at a crime scene
- Understand the difference between and location of visible and latent data
- List the areas of the computer that will be examined to retrieve forensic data

The BTK Killer

Dennis Rader was arrested in February 2005 and charged with committing ten murders since 1974 in the Wichita, Kansas, area. The killer, whose nickname stands for “bind, torture, kill,” hadn’t murdered since 1991, but resurfaced in early 2004 by sending a letter to a local newspaper taking credit for a 1986 slaying. Included with the letter were a photo-

copy of the victim's driver's license and three photos of her body. The BTK killer was back to his old habit of taunting the police. Three months later another letter surfaced. This time the letter detailed some of the events surrounding BTK's first murder victims. In 1974, he strangled Joseph and Julie Otero along with two of their children. Shortly after those murders occurred, BTK sent a letter to a local newspaper in which he gave himself the name BTK. In December 2004, a package found in a park contained the driver's license of another BTK victim along with a doll whose hands were bound with pantyhose and who was covered with a plastic bag.

The major break in the case came when BTK sent a message on a floppy disk to a local TV station. "Erased" information on the disk was recovered and restored by forensic computer specialists, and the disk was traced to the Christ Lutheran Church in Wichita. The disk was then quickly linked to Dennis Rader, the church council president. The long odyssey of the BTK killer was finally over.

Since the 1990s, few fields have progressed as rapidly as computer technology. Computers are no longer a luxury, nor are they in the hands of just a select few. Technology and electronic data are a part of everyday life and permeate all aspects of society. Consequently, computers have become increasingly important as sources of evidence in an ever-widening spectrum of criminal activities.

Investigators frequently encounter computers and other digital devices in all types of cases. As homicide investigators sift for clues they may inquire whether the method for a murder was researched on the Internet; whether signs of an extramarital affair can be found in e-mail or remnants of instant messages, which might provide motive for a spouse killing or murder for hire; or whether threats were communicated to the victim prior to a murder by an obsessed stalker. Arson investigators want to know whether financial records on a computer might provide a motive in an arson-for-profit fire. A burglary investigation would certainly be aided if law enforcement determined that the proceeds from a theft were being sold online—perhaps through eBay or a similar online auction site.

Accessibility to children and the perception of anonymity has given sexual predators a way to seek out child victims online. The vulnerability of computers to hacker attacks is a constant reminder of security issues surrounding digitally stored data. Finally, the fact that computers control most of our critical infrastructure makes technology an appetizing target for would-be terrorists. [100]

Computer forensics involves the preservation, acquisition, extraction, analysis, and interpretation of computer data. Although this is a simple definition, it gets a bit more complicated. Part of this complication arises from technology itself. More and more devices are capable of storing electronic data: cell phones, personal digital assistants (PDAs), iPods, digital cameras, flash memory cards, smart cards, jump drives, and many others. Methods for extracting data from these devices each present unique challenges. There are, however, sound forensic practices that apply to all these devices. The most logical place to start to examine these practices is with the most common form of electronic data: the personal computer.

FROM INPUT TO OUTPUT: HOW DOES THE COMPUTER WORK?

Hardware versus Software

Before we get into the nuts and bolts of computers, we must establish the important distinction between hardware and software. **Hardware** comprises the physical components of the computer: the computer chassis, monitor, keyboard, mouse, hard disk drive, random-access memory (RAM), and central processing unit (CPU), and so on (see Figure 17–1). The list is much more extensive, but generally speaking, if it is a computer component or peripheral that you can see, feel, and touch, it is hardware.

Software, conversely, is a set of instructions compiled into a program that performs a particular task. Software consists of programs and applications that carry out a set of instructions on the hardware. Operating systems (Windows, Mac OS, Linux, Unix), word-processing programs (Microsoft Word, WordPerfect), web-browsing applications (Internet Explorer, Netscape Navigator, Firefox), and accounting applications (Quicken, QuickBooks, Microsoft Money) are all examples of software. It is important not to confuse software with the physical media that it comes on. When you buy an application such as Microsoft Office, it comes on a compact disc (CD). The CD containing this suite of applications is typically referred to as software, but this is technically wrong. The CD is external computer media that contains the software; it is a container for and a medium to load the set of instructions onto the hard disk drive (the hardware).

Computer Case/Chassis

The case is the physical box holding the fixed internal computer components in place. Cases come in many shapes and sizes: a full upright tower chassis, a slim desktop model sitting on the desktop, or an all-in-one monitor/computer case like the iMac. For our purposes, the term *system unit* is probably most appropriate when describing a chassis seized as evidence. The term *system unit* accurately references the chassis, including the motherboard and other internal components.

Power Supply

The term *power supply* is actually a misnomer, because it doesn't actually supply power—the power company does that. Rather, a computer's power supply converts power from the wall outlet to a usable format for the computer and its components. Different power supplies have different wattage ratings. The use, or more specifically the components, of the computer dictate the appropriate power supply.

Motherboard

The main circuit board in a computer (or other electronic devices) is referred to as the **motherboard**. Motherboards contain sockets for chips and slots for add-on cards. Examples of add-on cards are a video card to connect the computer to the monitor, a network card or modem to connect to an internal network or the Internet, and a sound card to connect to speakers. Sockets on the motherboard typically accept things like random-access memory (RAM) or the central processing unit (CPU). The keyboard, mouse, CD-ROM drives, floppy disk drives, monitor, and other peripherals or components connect to the motherboard in some fashion through a direct wired or wireless connection.

System Bus

Contained on the motherboard, the system bus is a vast complex network of wires that carry data from one hardware device to another. This network is analogous to a complex highway. Data is sent along the bus in the form of ones and zeros (or, more appropriately stated, as electrical impulses representing an "on" or "off" state—this two-state computing is also known as *binary computing*).

Read-Only Memory (ROM)

This rather generic term describes special chips on the motherboard. ROM chips store programs called *firmware*, used to start the boot process and configure a computer's components. Today's ROM chips, termed *flash ROM*, are a combination of two types of chips used in past motherboard technologies. The first was known as the *system ROM*, which was responsible for booting the system and handling the "assumed" system hardware present in the computer. As the system ROM, generally speaking, could not be altered, and because as technology matured changes to the "assumed" hardware were more common, a different type of chip was introduced. The *complementary metal-oxide semiconductor* (CMOS) was a separate chip that allowed the user to exercise setup control over several system components. Regardless of how this technology is present on the motherboard, it can be referred to as the BIOS, for *basic input-output system*. The operation of the BIOS is relevant to several computer forensic procedures, particularly the boot sequence. It is the set of routines associated with the BIOS in ROM that initiates the booting process and enables the computer to communicate with various devices in the system such as disk drives, keyboard, monitor, and printer. As will become clear later, it is important not to boot the actual computer under investigation to the original hard disk drive. This would cause changes to the data, thus compromising the integrity of evidence. The BIOS allows investigators to control the boot process to some degree.

Central Processing Unit (CPU)

The **central processing unit (CPU)**, also referred to as a processor, is essentially the brain of the computer. It is the main (and typically the largest) chip that plugs into a socket on the motherboard. The CPU is the part of the computer that actually computes. Basically, all operations performed by the computer are run through the CPU. The CPU carries out the program steps to perform the requested task. That task can range from opening and working in a Microsoft Word document to performing advanced mathematical algorithms. CPUs come in various shapes, sizes, and types. Intel Pentium chips and Advanced Micro Devices (AMD) chips are among the most common.

Random-Access Memory (RAM)

This is one of the most widely mentioned types of computer memory. **Random-access memory (RAM)** takes the burden off the computer's processor and hard disk drive (HDD). If the computer had to access the HDD each time it wanted data, it would run slowly and inefficiently. Instead the computer, aware that it may need certain data at a moment's notice, stores the data in RAM. It is helpful to envision RAM as chips that create a large spreadsheet, with each cell representing a memory address that the CPU can use as a reference to retrieve data. RAM is referred to as *volatile memory* because it is not permanent; its contents undergo constant change and are forever lost once power is taken away from the computer. RAM takes the physical form of chips that plug into the motherboard; SIMMs (Single Inline Memory Modules), DIMMs (Dual Inline Memory Modules), and SDRAM (Synchronous Dynamic Random-Access Memory) are just a few of the types of chips. Today's computers come with various denominations of RAM: 256 MB (megabytes), 512 MB, and 1 GB (gigabyte) are the most common.[1]

Input Devices

Input devices are used to get data into the computer or to give the computer instructions. Input devices constitute part of the "user" side of the computer. Examples include the keyboard, mouse, joystick, and scanner.

Output Devices

Output devices are equipment through which data is obtained from the computer. Output devices are also part of the “user” side of the computer, and provide the results of the user’s tasks. They include the monitor, printer, and speakers.

Hard Disk Drive (HDD)

Generally speaking, the **hard disk drive (HDD)** is the primary component of storage in the personal computer (Figure 17–2). It typically stores the operating system (Windows, Mac OS, Linux, Unix), the programs (Microsoft Word, Internet Explorer, Open Office for Linux, and so on) and data files created by the user (documents, spreadsheets, accounting information, the company database, and so on). Unlike RAM, the HDD is permanent storage and retains its information even after the power is turned off. HDDs work off a controller that is typically part of the motherboard, but sometimes takes the form of an add-on (expansion) card plugged into the motherboard. The most common types of HDD controllers are integrated drive electronics (IDE), small computer system interface (SCSI), and serial ATA (SATA). Each HDD type has a different interface that connects it to the controller. Regardless of the type of controller, the data is basically stored in the same fashion. HDDs are mapped (formatted) and have a defined layout. They are logically divided into sectors, clusters, tracks, and cylinders. (See the section titled “How Data Is Stored” for further information).

Other Common Storage Devices

Although the HDD is the most common storage device for the personal computer, many others exist. Methods for storing data and the layout of that data can vary from device to device. A CD-ROM, for example, uses a different technology and format for writing data than a floppy disk or USB thumb drive. Fortunately, regardless of the differences among devices, the same basic forensic principals apply for acquiring the data. Common storage devices include the following:

CD-R/RW (compact disc-record/rewrite) and DVD-R/RW (DVD-record/rewrite). Compact discs (CDs) and digital video discs (DVDs) are two of the most common forms of storing external data. They are used to store everything from music and video to data files. A largely plastic disc with an aluminum layer is read by laser light in the CD/DVD reader. Different CDs are encoded in different ways, making the job of the forensic examiner difficult at times.

Floppy Disks. Though "floppies" are not as common as they once were, forensic examiners still encounter the 3.5-inch floppy disk. Floppy disks can be used to boot an operating system or to store data. They are constructed of hard plastic with a thin plastic disk on the inside. That thin plastic disk is coated with a magnetic iron oxide material. The disk is mapped and stores data in a similar fashion to the hard disk drive. By today's standards, floppy disks don't hold much data.

Zip Disks. Similar in structure to floppy disks, Zip disks hold a much larger amount of data. They come in several storage capacities, each with their own drive.

USB Thumb Drives and Smart Media Cards. These devices can store a large amount of data—some as much as 4 GB. They are known as solid-state storage devices because they have no moving parts. Smart media cards are typically found in digital cameras and PDAs, while USB thumb drives come in many shapes, sizes, and storage capacities.

Tapes. Tapes come in many different formats: 4 mm, 8 mm and storage capacities. Each typically comes with its own hardware reader and sometimes a proprietary application to read and write its contents. Tapes are typically used for backup purposes and consequently have great forensic potential.

Network Interface Card (NIC)

Very rarely do we find a computer today that doesn't have a NIC. Whether they are on a local network or the Internet, when computers need to communicate with each other, they typically do so through a NIC. NICs come in many different forms: add-on cards that plug into the motherboard, hard-wired devices on the motherboard, add-on cards (PCMCIA) for laptops, and universal serial bus (USB) plug-in cards, to name a few. Some are wired cards, meaning they need a physical wired connection to participate on the network, and others are wireless, meaning they receive their data via radio waves.

PUTTING IT ALL TOGETHER

A person approaches the computer, sits down, and presses the power button. The power supply wakes up and delivers power to the motherboard and all of the hardware connected to the computer. At this point the flash ROM chip on the motherboard (the one that contains the BIOS) conducts a power-on self test (POST) to make sure everything is working properly. The flash ROM also polls the motherboard to check the hardware that is attached and follows its programmed boot order, thus determining from what device it should boot. Typically the boot device is the HDD, but it can also be a floppy disk, CD, or USB drive. If it is the HDD, the HDD is then sent control. It locates the first sector of its disk (known as the master boot record), determines its layout (partition(s)), and boots an operating system (Windows, Mac OS, Linux, Unix). The person is then presented with a computer work environment, commonly referred to as a desktop. Now ready to work, the user double-clicks an icon on the desktop, such as a Microsoft Word shortcut, to open the program and begin to type a document. The CPU processes this request, locates the Microsoft Word program on the HDD (using a predefined map of the drive called a *file system table*), carries out the programming instructions associated with the application, loads Microsoft Word into RAM via the system bus, and sends the output to the monitor by way of the video controller, which is either located on or attached to the motherboard. The user then begins to type, placing the data from the keyboard into RAM. At the end, the user might print the document or simply save it to the HDD for later retrieval. If printed, the data is taken from RAM, processed by the CPU, placed in a format suitable for printing, and sent through the system bus to the external port where the printer is connected. If the document is saved, the data is taken from RAM, processed by the CPU, passed to the HDD controller (IDE, SCSI, or SATA) by way of the system bus, and written to a portion of the HDD. The HDD's file system table is updated so it knows where to retrieve that data later. In actuality, the boot process is more complex than the way it has been described above and requires the forensic examiner to possess an in-depth knowledge of its process.

The preceding example illustrates how three components perform the majority of the work: the CPU, RAM, and system bus. The example can get even more complicated as the user opens more applications and performs multiple tasks simultaneously (*multitasking*). Several tasks can be loaded into RAM at once and the CPU is capable of juggling them all. This allows for the multitasking environment and the ability to switch back and forth between applications. All of this is orchestrated by the operating system and is written in the language of the computer—ones and zeros. The only detail missing, and one that is important from a forensic standpoint, is a better understanding of how data is stored on the hard disk drive (see Figure 17–2).

HOW DATA IS STORED

Before beginning to understand how data is stored on a hard disk drive (HDD), it is first important to understand the role of the **operating system (OS)**. An OS, such as Windows, Mac OS, Linux, or Unix, is the bridge between the human user and the computer's electronic components.

It provides the user with a working environment and facilitates interaction with the system's components. Each OS supports certain types of file systems that store data in different ways, but some support the methods of others. Generally speaking, before an OS can write to a HDD it must first be formatted. But even before it can be formatted, a partition must be defined. A **partition** is nothing more than a contiguous set of blocks that are defined and treated as an independent disk. This means that a hard disk drive can hold several partitions, making a single HDD appear as several disks. Partitioning a drive can be thought of as dividing a container that begins as nothing more than four sides with empty space on the inside. We then cut a hole in the front of it and place inside two drawers containing the hardware to open and close them. We have just created a two-drawer filing cabinet and defined each drawer as contiguous blocks of storage. A partitioning utility such as Disk Manager or fdisk defines the drawer or drawers (partitions) that will later hold the data on the HDD. Just as the style, size, and shape of a filing cabinet drawer can vary, so too can partitions. After a hard drive is partitioned, it is typically formatted. (At this point it is a high-level format, not to be confused with low-level format, which is generally done by the manufacturer of the HDD.) The formating process initializes portions of the HDD and creates the structure of the file system. The file system can be thought of as the system for storing and locating data on a storage device; but more on this in a bit. Some of the file system types are FAT12 (typically on floppy disks), FAT16 (older DOS and older Windows partitions), FAT32 (Windows file systems), NTFS (most current Windows systems—2000 and XP), EXT2 and EXT3 (Linux systems), and HPFS (some Macintosh systems).

Each of these file systems has a different way of storing, retrieving, and allocating data. So, in summary, it can be said that a drive is prepared in three processes: low-level formatting (typically done by the manufacturer, dividing the platters into tracks and sectors), partitioning (accomplished through a utility such as fdisk or Disk Manager, defining a contiguous set of blocks), and formatting (initializing portions of the disk and creating the file system structure). Although a bit more technical and detailed, at the conclusion of these processes, the drive is logically defined. We say “logically” because no real divisions are made. If you were to crack open the HDD before or after partitioning and formatting, to the naked eye the platters would look the same. As shown in Figure 17–3, HDDs contain several platters stacked vertically which are logically divided into sectors, clusters, tracks, and cylinders. **Sectors** are typically 512 bytes in size (a **byte** is eight bits; a **bit** is a single one or zero). **Clusters** are groups of sectors; their size is defined by the file system, but they are always in sector multiples of two. (Although an NTFS partition does permit a one-sector-per-cluster scenario, such a scenario is not usually chosen.) A cluster, therefore, consists of two, four, six, or eight sectors, and so on. (With modern file systems, the user can exercise some control over the amount of sectors per cluster.) Tracks are concentric circles that are defined around the platter. Cylinders are groups of tracks that reside directly above and below each other. Additionally, the HDD has a file system table (map) of the layout of the defined space in that partition. FAT file systems use a *file allocation table* (which is where the acronym *FAT* comes from) to track the location of files and folders (data) on the HDD, while NTFS file systems (used by most current Windows systems—2000 and XP) use, among other things, a *master file table (MFT)*. Each file system table tracks data in different ways, and computer forensic examiners should be versed in the technical nuances of the HDDs they examine. It is sufficient for our purposes here, however, to merely visualize the file system table as a map to where the data is located. This map uses the numbering of sectors, clusters, tracks, and cylinders to keep track of the data.

One way to envision a partition and file system is as a room full of safe-deposit boxes. The room itself symbolizes the entire partition and the boxes symbolize clusters of data. In order to determine who rented which box, and subsequently where their property is, a central database is needed. This is especially true if a person rented two boxes located in opposite ends of the room (noncontiguous data on the HDD). The database tracking the locations of the safe-deposit boxes is much like a file system table tracking the location of data within the clusters. This example is also useful to understand the concept of reformatting a HDD. If the database managing the locations of the safe-deposit boxes were wiped out, the property in them would still remain; we just wouldn't know what was where. So too with the hard disk drive. If a user were to wipe the file system table clean—for example, by reformatting it—the data itself would not be gone. Both the database tracking the locations of the safe-deposit boxes and the file system table tracking the location of the data in the cluster are maps—not the actual contents. (Exceptions exist with some file systems, such as an NTFS file system, which stores data for very small files right in its file system table, known as the master file table).

PROCESSING THE ELECTRONIC CRIME SCENE

Processing the electronic crime scene has a lot in common with processing a traditional crime scene. The investigator must first ensure that the proper legal requirements (search warrant, consent, and so on) have been met so that the scene can be searched and the evidence seized. The investigator should then devise a plan of approach based on the facts of the case and the physical location. The scene should be documented in as much detail as possible before disturbing any evidence, and before the investigator lays a finger on any computer components. Of course there are circumstances in which an investigator might have to act quickly and pull a plug before documenting the scene, such as when data is in the process of being deleted.

Crime-scene documentation is accomplished through two actions: sketching and photographing. The electronic crime scene is no different. The scene should be sketched in a floor plan fashion (see Figure 17–4) and then overall photographs of the location taken. In the case of a network, a technical network sketch should also be included if possible. After taking photographs of the overall layout, close-up photographs should be shot. A close-up photograph of any running computer monitor should be taken. All the connections to the main system unit, such as peripheral devices (keyboard, monitor, speakers, mouse, and so on), should be photographed. If necessary, system units should be moved delicately and carefully to facilitate the connections photograph. (See Figure 17–5a). Close-up photographs of equipment serial numbers should be taken if practical.

At this point, investigators must decide whether to perform a live acquisition of the data, perform a system shutdown (as in the case of server equipment), pull the plug from the back of the computer,[2] or a combination thereof. Several factors influence this decision. For example, if encryption is being used and by pulling the plug the data will encrypt, rendering it unreadable without a password or key, pulling the plug would not be prudent. Similarly, if crucial evidentiary data exists in RAM and has not been saved to the HDD, the data will be lost. Hence, if power to the system is discontinued another option must be considered. Regardless, the equipment will most likely be seized. Exceptions exist in the corporate environment, where servers are fundamental to business operations.

After the photographs and sketches are complete, but before disconnecting the peripherals from the computer, a label should be placed on the cord of each peripheral, with a corresponding label placed on the port to which it is connected. A numbering scheme should be devised to identify the system unit if several computers are at the scene (Figure 17–5b). The combination of sketching, photographing, and labeling should adequately document the scene, prevent confusion of which component went with which system unit, and facilitate reconstruction if necessary for lab or courtroom purposes.

Forensic Image Acquisition

Now that the items have been seized, the data needs to be obtained for analysis. The number of electronic items that potentially store evidentiary data are too vast to cover in this section. The hard disk drive will be used as an example, but the same "best practices" principles apply for other electronic devices as well.

Throughout the entire process, the computer forensic examiner must use the least intrusive method. The goal in obtaining data from a HDD is to do so without altering even one bit of data. Because booting a HDD to its operating system changes many files and could potentially destroy evidentiary data, obtaining data is generally accomplished by removing the HDD from the system and placing it in a laboratory forensic computer so that a forensic image can be created. However, the BIOS of the seized computer sometimes interprets the geometry of the HDD differently than the forensic computer does. In these instances, the image of the HDD must be obtained using the seized computer. Regardless, the examiner must ensure that the drive to be analyzed is in a "write-blocked," read-only state when creating the forensic image. Furthermore, the examiner needs to be able to prove that the forensic image he or she obtained includes every bit of data and caused no changes (writes) to the HDD. To this end, a sort of fingerprint of the drive is taken before and after imaging. This fingerprint is taken through the use of a **Message Digest 5 (MD5)**, **Secure Hash Algorithm (SHA)**, or similar validated algorithm. Before imaging the drive the algorithm is run and a 32-character alphanumeric string is produced based on the drive’s contents. The algorithm is then run against the resulting forensic image; if nothing changed, the same alphanumeric string is produced, thus demonstrating that the image is all-inclusive of the original contents and that nothing was altered in the process.

A forensic image of the data on a HDD (and the same holds true for floppy disks, CDs, DVDs, tapes, flash memory devices, and any other storage medium) is merely an exact duplicate of the entire contents of the drive. In other words, all portions of the drive are copied from the first bit (one or zero) to the last. Why would investigators want to copy what appears to be blank or unused portions of the HDD? The answer is simple: to preserve latent data, discussed later in the chapter. It suffices to say here that data exists in areas of the drive that are, generally speaking, unknown and inaccessible to most end users. This data can be valuable as evidence. Therefore, a forensic image—one that copies every single bit of information on the drive—is necessary.[3] A forensic image differs from a backup or standard copy in that it takes the entire contents, not only data the operating system is aware of.

Many forensic software packages come equipped with a method to obtain the forensic image. The most popular software forensic tools—EnCase, Forensic Toolkit (FTK), Forensic Autopsy (Linux-based freeware), and SMART (Linux-based software by ASR Data)—all include a method to obtain a forensic image. All produce self-contained image files that can then be interpreted and analyzed. They also allow image compression to conserve storage. The fact that self-contained, compressed files are the result of forensic imaging allows many images from different cases to be stored on the same forensic storage drive. This makes case management and storage much easier (see Figure 17–6).

Analysis of Electronic Data

Analysis of electronic data is virtually limitless and bound only to the level of skill of the examiner. The more familiar an examiner is with computers, operating systems, application software, data storage, and a host of other disciplines, the more prepared he or she will be to look for evidentiary data. Because computers are vast and complex, discussing each area, file, directory, log, or computer process that could potentially contain evidentiary data is beyond the scope of one chapter—and may be beyond the scope of an entire book. What follows are some of the more common areas of analysis. While reading this section, reflect on your own knowledge of computers and consider what other data might be of evidentiary value and where it might be found.

EVIDENTIARY DATA

Visible Data

The category of **visible data** includes all information that the operating system is presently aware of, and thus is readily accessible to the user. Here we present several common types of visible data considered in many investigations. This list is by no means exhaustive and can include any information that has value as evidence.

Data/Work Product Files. One place to find evidence is in documents or files produced by the suspect. This category is extremely broad and can include data from just about any software program. Microsoft Word and WordPerfect word-processing programs typically produce text-based files such as typed documents and correspondence. These programs, and a host of other word-processing programs, have replaced the typewriter. They are common sources of evidence in criminal cases, particularly those involving white-collar crime.

Also relevant in white-collar crime and similar financial investigations are any data related to personal and business finance. Programs such as QuickBooks and Peachtree accounting packages can run the entire financial portion of a small to midsize business. Similarly, it is not uncommon to find personal bank account records in the computer that are managed with personal finance software such as Microsoft Money and Quicken. Moreover, criminals sometimes use these programs as well as spreadsheet applications to track bank accounts stolen from unsuspecting victims. Computer forensic examiners should familiarize themselves with these programs, the ways in which they store data, and methods for extracting and reading the data.

Advances in printer technology have made high-quality color printing both affordable and common in many homes. While this is a huge benefit for home office workers and those interested in graphic arts, the technology has been used for criminal gain. Counterfeiting and check and document fraud are easily perpetrated by most home computer users. All that is required is a decent ink-jet printer and a scanner. Including the computer, a criminal could set up a counterfeiting operation for less than $1500. Examiners must learn the graphics and photo-editing applications used for nefarious purposes. Being able to recognize the data produced by these applications and knowing how to display the images is key to identifying the evidence.

Swap File Data. When an application is running, the program and the data being accessed are loaded into RAM. A computer's RAM is much faster than the "read" speed of the hard disk drive, and that's why the programs are loaded here—for fast access and functioning. RAM, however, has its limits. Some computers have 256 MB of RAM, others 512 MB, and still others as much as a gigabyte or two. Regardless of the amount, though, most operating systems (Windows, Linux, and so on) are programmed to conserve RAM when possible. This is where the **swap file** comes in. The operating system attempts to keep only data and applications that are presently being used in RAM. Other applications that were started, but are currently waiting for user attention, may be swapped out of RAM and written to the swap space on the hard disk drive.[4] For example, a manager of a retail store may want to type a quarterly report based on sales. The manager starts Microsoft Word and begins his report. Needing to incorporate sales figure data from a particular spreadsheet, he opens Microsoft Excel. Depending on what is running on the computer, the original Word document may be swapped from RAM to the swap space on the HDD to free up space for Excel. As the manager goes back and forth between the programs (and maybe checks his e-mail in between) this swapping continues. Data that is swapped back and forth is sometimes left behind in the swap space. Even as this area is constantly changed, some of the data is orphaned in unallocated space, an area of the HDD discussed later in this chapter.

Swap file or space can be defined as a particular file or even a separate HDD partition, depending on the operating system and file system type (FAT, NTFS, EXT2, and so on). For Windows systems either the swap file *Win386.sys* or *pagefile.sys* is used, depending on the specific Windows version and file system type. Linux systems can create partitions just for swapping data in and out of RAM. Data in the swap space can be read by examining the HDD through forensic software or a utility that provides a binary view, such as Norton Disk Editor or WinHex (see Figure 17–7).

Temporary Files. Any user who has suffered a sudden loss of power in the middle of typing a document can attest to the value of a **temporary file**. Most programs automatically save a copy of the file being worked on in a temporary file. After typing a document, working on a spreadsheet, or working on a slide presentation, the user can save the changes, thus promoting the temporary copy to an actual file. This is done as a sort of backup on the fly. If the computer experiences a sudden loss of power or other catastrophic failure, the temporary file can be recovered, limiting the amount of data lost. The loss is limited because the temporary file is not updated in real time. Rather, it is updated periodically (typically defaulted to every ten minutes in most programs), depending on the application's settings. Temporary files can sometimes be recovered during a forensic examination. Additionally, some of the data that may have been orphaned from a previous version may be recoverable, if not the complete file. This is true even when a document has been typed and printed, but never saved. The creation of the temporary file makes it possible for some of this "unsaved" data to be recovered during analysis.

Another type of temporary file valuable to the computer investigator is the print spool file. When a print job is sent to the printer a spooling process delays the sending of the data to the printer. This happens so the application can continue to work while the printing takes place in the background. To facilitate this, a temporary print spool file is created; this file typically includes the data to be printed and information specific to the printer. There are different methods for accomplishing this, and thus the files created as a result of this process vary. It is sometimes possible to view the data in a readable format from the files created during the spooling process.

Latent Data

The term **latent data** includes data that are obfuscated (not necessarily intentionally) from a user's view. It includes areas of files and disks that are typically not apparent to the computer user, but contain data nonetheless. Latent data are one of the reasons a forensic image of the media is created. If a standard copy were all that was produced, only the logical data (that which the operating system is aware of) would be captured. Getting every bit of data ensures that potentially valuable evidence in latent data is not missed.

Once the all-inclusive forensic image is produced, how is the latent data viewed? Utilities that allow a user to examine a hard disk drive on a binary (ones and zeros) level are the answer.

Applications such as Norton Disk Editor and WinHex provide this type of access to a hard disk drive or other computer media. Thus these applications, sometimes also referred to as *hex editors* (for the hexadecimal shorthand of computer language), allow all data to be read on the binary level independent of the operating system's file system table. Utilities such as these can write to the media under examination, thus changing data. Consequently, a software or hardware write-blocker should be used. A more common option in data forensics is to use specialized forensic examination software. EnCase and Forensic Toolkit for Windows and SMART and Forensic Autopsy for Linux are examples of forensic software. Each allows a search for evidence on the binary level and provides automated tools for performing common forensic processing techniques. Examiners should be cautious, however, about relying too heavily on automated tools. To merely use an automated tool without understanding what is happening in the background and why evidentiary data might exist in particular locations would severely impede the ability to testify to the findings.

Slack Space. Slack space can really be divided into two separate areas—**file slack** and **RAM slack**. Before we can begin to understand the concept of either, we must return to how files are stored. To illustrate this concept it is best to use the example of a simple Windows partition. Recall that a partition system is nothing more than a contiguous set of blocks that are defined and treated as an independent disk. Remember that although the smallest unit of data measure is one bit (either a one or a zero), a HDD cannot address or deal with such a small unit. In fact, not even a byte (eight bits) can be addressed. Rather, the smallest unit of addressable space by a HDD is the sector. Sectors are groups of bytes and can vary in size depending on the media; HDDs typically group sectors in 512-byte increments, while CD-ROMs allocate 2048 bytes per sector. Even though the sector is the smallest addressable unit by the HDD, the operating system and the file system on the HDD view it a bit differently. File systems may mandate a minimum amount of space allocated to each file. This returns us to the concept of clusters.

As you may recall, clusters are groups of sectors used to store files and folders. The cluster is the minimum storage unit defined and used by the logical partition. These clusters are maintained by the tables or bitmaps of the file system. (Remember, the tables and bitmaps—FAT, NTFS, EXT2—are similar to databases that track safe-deposit boxes, letting us know where to find things.) It is because of the minimum addressable sector of the HDD and the minimum unit of storage requirement of the volume that we have slack space.

If the minimum addressable unit of the HDD is 512 bytes, what happens if the file is only 100 bytes? In this instance there are 412 bytes of slack space. It does not end here, however, because of the minimum cluster requirement. Minimum cluster allocation must be defined in a sector multiple of two. Thus a cluster must be a minimum of two, four, six, or eight sectors, and so on. So, if we return to our initial example of the 100-byte file and apply it to a two-sector-per-cluster (1024 bytes) volume requirement, we now isolate 1024 bytes (two sectors) of storage space for a 100-byte file. The remaining 924 bytes would be slack space (see Figure 17–8).

To illustrate this point, let us expand on the concept of safe-deposit boxes. The bank offers safe-deposit boxes of a particular size. This is the equivalent of the HDD’s clusters. A person wanting to place only a deed to a house in the box gets the same size box as a person who wants to stuff it full of cash. The former would have empty space should he or she desire to place additional items in the box. This empty space is the equivalent of slack space. But what if the box becomes full and the person needs more space? That person must then get a second box. Similarly, if a file grows to fill one cluster and beyond, a second cluster (and subsequent clusters as needed) is allocated. The remaining space in the second cluster is slack space. This continues as more and more clusters are allocated depending on file size and file growth.

This example is a bit of an oversimplification because there are actually two types of slack space: RAM slack and file slack. **RAM slack** occupies the space from where the actual (logical) data portion of the file ends to where the first allocated sector in the cluster terminates. **File slack**, therefore, occupies the remaining space of the cluster. Let us go back to the 100-byte file with the two-sector-per-cluster minimum requirement. Following the end of the logical data (the end of the 100 bytes), the remaining 412 bytes of that sector is RAM slack; the additional 512 bytes completing the cluster is then file slack. See Figure 17–9 for a visual depiction. The question now becomes: What can I expect to find in slack space and why is this important? The answer: junk—valuable junk.

RAM slack is a concept that was more relevant in older operating systems. Remember that the minimum amount of space the HDD can address is the 512-byte sector. Therefore if the file size is only 100 bytes, the remaining space must be padded. Some operating systems pad this area with data contained in RAM. This could include Web pages, passwords, data files, or other data that existed in RAM when the file was written. Modern Windows operating systems pad this space with zeros, but some examinations may still yield valuable data in this area.

File slack, on the other hand, can contain a lot of old, orphaned data. To illustrate this point, let's take the 100-byte file example a bit further. Let's say that prior to the 100-byte file being written to the HDD and occupying one cluster (two sectors totaling 1024 bytes), a 1,000-byte file occupied this space but was deleted by the user. Understanding that when a file is "deleted" the data still remains behind, so it is probably a safe bet that data from the original 1000-byte file remains in the slack space of the new 100-byte file now occupying this cluster. This is just one example of why data exists in file slack and why it might be valuable as evidence.

In one final attempt to illustrate this point, let us again build on our safe-deposit box analogy. If a person rents two safe-deposit boxes—each box representing a sector and combined representing a cluster—and that person places the deed to his house in the first box, the remaining space of that first box would be analogous to RAM slack. The space in the second box would be the equivalent of file slack. The only difference is that unlike the empty spaces of the safe-deposit box, the slack space of the file most likely contains data that might be valuable as evidence.

The data contained in RAM and file slack is not really the concern of the operating system. As far as the OS is concerned, this space is empty and therefore ready to be used. Until that happens, however, an examination with one of the aforementioned tools will allow a look into these areas, thus revealing the orphaned data. The same is true for unallocated space.

Unallocated Space. Latent evidentiary data also resides in **unallocated space**. What is unallocated space, how does data get in there, and what is done to access this space? If we have a 80 GB hard drive and only half of the hard drive is filled with data, then the other half, or 40 GB, is unallocated space (see Figure 17–10.) Returning to our safe-deposit box analogy, if the entire bank of safe-deposit boxes contains 100 boxes, but only 50 are currently in use, then the other 50 would be the equivalent of unallocated space. The HDD’s unallocated space typically contains a lot of useful data. The constant shuffling of files on the HDD causes data to become orphaned in unallocated space as the logical portion of the file is rewritten to other places. Some examples of how data is orphaned may help.

Defragmenting. Defragmenting a HDD involves moving noncontiguous data back together. Remember that the HDD has minimum space reservation requirements. Again, if the file requires only 100 bytes of space, the operating system might allocate much more than that for use. If the file grows past what has been allocated for it, another cluster is required. If, however, a different file occupies the next cluster in line, then the operating system will have to find another place for that first file on the drive. In this scenario, the file is said to be *fragmented* because data for the same file is contained in noncontiguous clusters. In the case of the HDD, the shuffling of files causes data to be orphaned in unallocated space. Ultimately fragmentation of numerous files can degrade the performance of a HDD, causing the read/write heads to have to traverse the platters to locate the data. Defragmenting the HDD takes noncontiguous data and rearranges it so it is in contiguous clusters. Building yet again on our safe-deposit box analogy, if a renter eventually needs to store more property than his original box can hold, the bank will rent him a second box. If, however, all the boxes around his are occupied and the only free one is in another section of the room, then his property is "fragmented." The bank would have to "defrag" the safe-deposit boxes to get the property of users with more than one box into adjacent boxes.

Swap File/Swap Space. Recall that a computer uses the HDD to maximize the amount of RAM by constantly swapping data in and out of RAM to a predetermined location on the HDD, thus freeing valuable RAM. The constant read and write operations of RAM cause a constant change in the swap file—*WIN386.swp* or *pagefile.sys* in Windows—or swap space on a Linux system. Data can become orphaned in unallocated space from this constant swapping to and from the HDD.

Deleted Files. The deletion of files is another way that data becomes orphaned in unallocated space. Data from deleted files can manifest itself in different ways during a forensic examination. The actions that occur when a file is deleted vary among file systems. What is fairly consistent, though, is that generally speaking the data is not gone. For example, consider what happens when a user or program deletes a file in a Windows operating system with a FAT file system.

When a file is deleted the first character in the file's directory entry (its name) is replaced with the Greek letter sigma. When the sigma replaces the first character, the file is no longer viewable through conventional methods and the operating system views the space previously occupied by the file as available. The data, however, is still there.

This example doesn't account for the actions of the Windows Recycle Bin. When the Windows operating system is set up to merely place the deleted file in the Recycle Bin, the original directory entry is deleted and one is created in the Recycle folder for that particular user. The new Recycle folder entry is linked to another file, the *info* or *info2* file, which includes some additional data, such as the location of the file prior to its deletion should the user wish to restore it to that location. Detailed discussions of the function of the Recycle Bin are beyond the scope of this chapter, but suffice it to say that even when the Recycle Bin is emptied the data usually remains behind until overwritten. Moreover, Windows NTFS partitions and Linux EXT partitions handle deleted files differently, but in both cases data typically remains.

What if a new file writes data to the location of the original file? Generally speaking, the data is gone. This is, of course, unless the new file only partially overwrites the original. In this instance we return to the unallocated space orphaned data scenario: If a file that occupied two clusters is deleted, and a new file overwrites one of the clusters, then the data in the second cluster is orphaned in unallocated space. Of course yet a third file can overwrite the second cluster entirely, but until then the data remains in unallocated space. Let us once again look to our safe-deposit box analogy. If, for example, the owner of two safe-deposit boxes stopped renting them, the bank would list them as available. If the owner didn't clean them out, the contents would remain unchanged. If a new owner rented one of the boxes, the contents from the former owner would be replaced with the new owner's possessions. The second box would therefore still contain orphaned contents from its previous owner. The contents would remain in this “unallocated box” space until another renter occupies it.

Chapter Summary

Computers have permeated society and are used in countless ways with innumerable applications. Similarly, the role of electronic data in investigative work has realized exponential growth in the last decade. Users of computers and other electronic data storage devices leave footprints and data trails behind. Computer forensics involves the preservation, acquisition, extraction, analysis, and interpretation of computer data. In today’s world of technology, many devices are capable of storing data and could thus be grouped into the field of computer forensics.

The central processing unit (CPU) is the brain of the computer—the main chip responsible for doing the actual computing. Random-access memory (RAM) is volatile memory containing data that is forever lost when the power is turned off. Programs are loaded into RAM because of its faster read speed. The hard disk drive (HDD) is typically the primary location of data storage within the computer. Different operating systems map out HDDs differently and examiners must be familiar with the file system they are examining. Evidence exists in many different locations and in numerous forms on a HDD. This evidence can be grouped into two major categories: visible and latent data.

Visible data is data that the operating system is aware of, and consequently is easily accessible to the user. From an evidentiary standpoint, it can encompass any type of user-created data, such as word-processing documents, spreadsheets, accounting records, databases, and pictures. Temporary files, created by programs as a sort of backup on the fly, can also prove valuable as evidence. Finally, data in the swap space (used to conserve the valuable RAM within the computer system) can yield evidentiary visible data.

Latent data, on the other hand, is data that the operating system typically is not aware of. Evidentiary latent data can exist in both RAM slack and file slack. RAM slack is the area from the end of the logical file to the end of the sector. File slack is the remaining area from the end of the final sector containing logical data to the end of the cluster. Another area where latent data might be found is in unallocated space. Unallocated space is space on a HDD that the operating system sees as empty and ready for data. The constant shuffling of data through deletion, defragmentation, and swapping is one of the ways data is orphaned in latent areas. Finally, when a user deletes files the data typically remains behind. Deleted files are therefore another source of latent data to be examined during forensic analysis.

Computer file systems and data structures are vast and complex. Therefore, areas of forensic analysis are almost limitless and constrained only by the knowledge and skill of the examiner. With a working knowledge of a computer's function, how they are utilized, and how they store data, an examiner is on his or her way to begin to locate the evidentiary data.

Review Questions

1. Computer forensics involves the ___________, ___________, ___________, and ___________ of computer data.

2. True or False: Hardware comprises the physical components of the computer. ___________

3. ___________ is a set of instructions compiled into a program that performs a particular task.

4. (ROM, RAM) chips store programs used to start the boot process.

5. The term used to describe the chassis, including the motherboard and any other internal com-

ponents of a personal computer, is ___________.

6. True or False: The motherboard is a complex network of wires that carry data from one hardware device to another. ___________

7. True or False: The first thing you should do when you encounter a computer system in a forensic investigation is to connect the power supply and boot the system. ___________

8. RAM is referred to as volatile memory because it is not ___________.

9. The brain of the computer is referred to as the ___________.

10. The ___________ is the primary component of storage in the personal computer.

11. Personal computers typically communicate with each other through a(n) ___________.

12. The computer's ___________ permits the user to manage files and applications.

13. A hard drive's partitions are typically divided into ___________, ___________, ___________, and ___________.

14. A(n) ___________ is a single one or zero in the binary system, and the smallest term in the language of computers.

15. A(n) ___________ is a group of eight bits.

16. A group of sectors, always units in multiples of two, is called a(n) ___________.

17. An exact duplicate of the entire contents of a hard disk drive is known as a(n) ___________.

18. All data readily available to a computer user is known as ___________ data.

19. A(n) ___________ file is created when data is moved from RAM to the hard disk drive to conserve space.

20. Most programs automatically save a copy of a file being worked on into a(n) ___________ file.

21. The existence of ___________ data is why a forensic image of the media is created.

22. The smallest unit of addressable space on a hard disk drive is the ___________.

23. The two types of slack space are ___________ slack and ___________ slack.

24. ___________ slack is the area from the end of the logical to the end of the sector.

25. The portion of a disk that does not contain stored data is called ___________.

26. True or False: Defragmenting a hard disk drive involves moving noncontiguous data back together. ___________

27. True or False: A portion of a "deleted" file may be found in a computer's unallocated space. ___________

Further References

Britz, M. T., *Computer Forensics and Cyber Crime*. Upper Saddle River, N.J.: Prentice Hall, 2004.

Casey, E., *Digital Evidence and Computer Crimes*, 2nd ed., San Diego: Elsevier Academic Press, 2004.

Kruse, W. G., and J. G. Heiser, *Computer Forensics—Incident Response Essentials*. Boston: Addison-Wesley, 2001.

Nelson, B., A. Phillips, F. Enfinger, and C. Steuart, *Guide to Computer Forensics and Investigations*, 2nd ed. Boston: Thomson Course Technology, 2005.

Case Study 1

Computer Forensic Analysis Answers the Question "Arson or Accident?”

Brief

The home of John Smith was destroyed by a fire, which was later determined not to be an accident, but rather the result of arson. During the fire Smith’s wife, Jane, died. Investigators learn that insurance policies taken against both the home and the life of Jane Smith were recently increased. Smith stands to receive a very large monetary settlement. This fact, and problems with his purported alibi at the time of the fire, makes him the primary suspect. Smith has steadfastly denied the existence of the insurance policies and offers that his wife must have recently changed the policies. Further investigation discloses that the couple did not possess a home computer but that Smith uses a computer at work. After applying for and receiving a search warrant for Smith’s workplace, the arson investigator seizes the computer system unit from underneath Smith’s desk, which he found in a powered-off condition.

Furthermore, during the execution of the search warrant, the company’s computer administrator tells investigators that the computer was used only by Smith. The computer system unit is submitted for forensic analysis.

Analysis Request

Locate any incriminating or exculpatory evidentiary data with respect to Smith’s knowledge of changes in his insurance policy. Locate any evidentiary data with respect to motive for the crimes of arson and/or homicide.

Forensic Image Acquisition

1. The computer system was documented and its chassis was opened and a single IDE/ATA hard disk drive (HDD) was located and documented. The HDD was removed from the system and the computer system unit was booted to the BIOS setup program. The system date and time were verified.

2. The HDD was then placed in a forensic workstation, connected to the system using a hardware write-blocking device to ensure that the suspect HDD was not altered in any way.

3. A forensic image of the HDD was acquired using EnCase Version 5. The integrity of this image was verified using the MD5 algorithm inherent in the EnCase program. (A date and time analysis was done on all the files, revealing no dates later than the date of the execution of the search warrant).

Analysis

1. Deleted files were recovered.

2. All files, including dates and times, logical and physical sizes, and complete location path, were documented by the EnCase program.

3. User accounts were documented: two default accounts (*Administrator* and *Guest*, SID 500 and 501 respectively) and one user account (*jsmith*).

4. The operating system and file system type were documented: Windows 2000 using an NTFS partition.

5. Keyword text searches, derived from the text of letters received by the insurance company, were conducted.

6. All Microsoft Word documents (.doc and .rtf) were examined.

7. All text documents were examined.

8. Print spool files were examined.

Findings

1. A file titled *insurance1.doc* was located in the directory *C:\Documents and Settings\jsmith\junk*. The directory structure *C:\Documents and Settings\jsmith* coincides with a default directory for the user name *jsmith*, which would have been established with that account. The subdirectory *junk* was then added by a user of the *jsmith* account. The text in this file is the same as the one received by the insurance company requesting an increase in homeowner's insurance.

2. Text found in unallocated space matches sections of text in a second letter received by the insurance company requesting an increase in life insurance for Jane Smith.

3. A file titled ~ *WRL1604.tmp* was found in the directory *C:\Documents and Settings\jsmith\junk*. The file matches that of a temporary Microsoft Word file and contains text matching sections of text in the letter received by the insurance company requesting an increase in life insurance for Jane Smith.

4. A file titled *46127a.SPL* was located in the directory *C:\windows\ system32\spool\printers.* This file appears to be a print spool file. This file, when viewed as an Enhanced Meta File (EMF), revealed a document exact in composition and similar in layout to the letter received by the insurance company requesting an increase in life insurance for Jane Smith. An EMF is a type of spool file created during the printing process and can be viewed as a Windows picture file in the forensic software.

Conclusion

Based on the forensic examination of the computer data submitted in this case, it can be stated within a reasonable degree of scientific certainty that a user of this computer had knowledgeable interaction with letters very similar in content, composition, and structure to the evidentiary letters submitted as reference for analysis.

Case Study 2

Counterfeiting and Fraud: A Forensic Computer Investigation

Brief

A detective submits a computer laptop for examination and explains that it was seized in connection with a case of counterfeiting and fraud. According to the detective, patrol officers happened upon a large sport-utility vehicle, occupied by one male driver, parked in the lot of a local mall. According to the officers, the driver and the circumstances appeared suspicious. After investigating further, the officers located a laptop computer, color printer, and scanner in the rear of the vehicle. All equipment was hooked up and running. Additionally, the officers located gift certificates for one of the stores within the mall, which apparently were printed inside the vehicle. Finally, two $100 bills bearing exactly the same serial number were located in the driver’s wallet. In response to questioning, the driver admitted using the system to print bogus gift certificates and counterfeit cash, which he then redeemed inside the mall. Prior to submission at the computer forensics laboratory, the equipment was processed for fingerprints at the state Bureau of Criminal Identification (BCI).

Analysis Request

Locate any evidentiary data with respect to the crimes of counterfeiting and fraud. Demonstrate any connection between the recovered printed documents and the electronic equipment seized from the vehicle.

Forensic Image Acquisition

1. The computer system was documented and its case was opened and a single IDE hard disk drive (HDD) was located and documented. The HDD was removed from the system and the computer system unit was booted to the BIOS setup program. The system date and time were verified.
2. The HDD was then placed in a forensic workstation, connected to the system using a hardware write-blocking device to ensure that the suspect HDD was not altered in any way.
3. A forensic image of the HDD was acquired using EnCase Version 5. The integrity of this image was verified using the MD5 algorithm inherent in the EnCase program. A date and time analysis was done on all the files, revealing no dates later than the date of the execution of the search warrant.

Analysis

1. Deleted files were recovered.
2. All files, including dates and times, logical and physical sizes, and complete location path, were documented by the EnCase program.
3. The operating system and file system type were documented: Windows XP using an NTFS file system.
4. All graphics files were viewed, including ones previously deleted.
5. A graphics finder script was run against unallocated space. The script searched this area to locate file signatures of known graphics files.

6. All print spool files were located and examined.

Findings

1. A file titled *100front.jpg* was located in the directory *C:\Documents and Settings\user1\My Documents.* This file is an image of the front of a $100 bill. The serial number on this image matched the serial number of the suspected counterfeit $100 bills found on the suspect.

2. A file titled *100back.jpg* was located in the directory *C:\Documents and Settings\user1\My Documents.* This file is an image of the back of a $100 bill.

3. A file titled *GapGiftCert1.jpg* was located in the directory *C:\Documents and Settings\user1\My Documents.* This file is an image of the front of a gift certificate for The Gap, a retail store.

4. A file titled *GapGiftCert2.jpg* was located in the directory *C:\Documents and Settings\user1\My Documents.* This file is an image of the back of a gift certificate for The Gap, a retail store.

5. A file titled *thumbs.db* was located in the directory *C:\Documents and Settings\user1\My Documents.* This file, when viewed as a compound file, displayed several images, namely the images in items 1–4.

6. In the folder *C:\Documents and Settings\User1\My Recent Documents,* link files were found to the following:

   a. *C:\Documents and Settings\user1\My Documents\100front.jpg*

   b. *C:\Documents and Settings\user1\My Documents\100back.jpg*

   c. *C:\Documents and Settings\user1\My Documents\GapGiftCert1.jpg*

   d. *C:\Documents and Settings\user1\My Documents\GapGiftCert2.jpg*

7. The submitted scanner and printer were connected to a laboratory computer system and the aforementioned evidentiary files were copied onto the HDD of that system. Several printouts of the images were made. Additionally, test items were scanned and printed. All exemplars produced from the laboratory computer system were submitted to the state Bureau of Criminal Identification. The original counterfeit currency and gift certificates were also submitted to BCI for comparison to the exemplars. BCI was asked to locate any distinguishing characteristics produced by the printer and scanner submitted in this case.

Conclusion

Based on the forensic examination of the computer data submitted in this case, it can be stated within a reasonable degree of scientific certainty that a user of this computer knowingly produced counterfeit currency and counterfeit gift certificates.

Hardware

The physical components of a computer: case, keyboard, monitor, motherboard, RAM, HDD, mouse, and so on. Generally speaking, if it is a computer component you can touch, it is hardware.

Software

A set of instructions compiled into a program that performs a particular task. Software consists of programs and applications that carry out a set of instructions on the hardware.

Motherboard

The main system board of a computer (and many other electronic devices). It delivers power, data, and instructions to the computer's components. Every component in the computer connects to the motherboard, either directly or indirectly.

Central Processing Unit (CPU)

The main chip within the computer; also referred to as the brain of the computer. This microprocessor chip handles most of the operations (code and instructions) of the computer.

Random-Access Memory (RAM)

The volatile memory of the computer, when power is turned off, its contents are lost. Programs and instructions are loaded into RAM while they are in use.

Hard Disk Drive (HDD)

Typically the main storage location within the computer. It consists of magnetic platters contained in a case (usually 3.5″ in a desktop computer and 2.5″ in a laptop). The HDD is usually where the operating system, applications, and user data are stored.

Operating System (OS)

The software that provides the bridge between the system hardware and the user. The OS lets the user interact with the hardware and manages the file system and applications. Some examples are Windows (XP, 2000), Linux, and Mac OS.

Partition

A contiguous set of blocks that are defined and treated as an independent disk.

Sector

The smallest addressable unit of data by a hard disk drive; generally consists of 512 bytes.

Byte

A group of eight bits.

Bit

Short for *binary digit*; Taking the form of either a one or a zero, it is the smallest unit of information on a machine.

Cluster

A group of sectors in multiples of two. Cluster size varies from file system to file system and is typically the minimum space allocated to a file.

Message Digest 5 (MD5)/Secure Hash Algorithm (SHA)

A software algorithm used to “fingerprint” a file or contents of a disk; used to verify the integrity of data. In forensic analysis it is typically used to verify that an acquired image of suspect data was not altered during the process of imaging.

Visible Data

All data that the operating system is presently aware of, and thus is readily accessible to the user.

Swap File

A file or defined space on the HDD used to conserve RAM. Data is swapped (paged) to this file/space to free RAM for applications that are in use.

Temporary Files

Files temporarily written by an application to perform a function. For applications, such as Microsoft Word and Excel, temporary files are created to provide a “backup” copy of the work product should the computer experience a catastrophic failure.

Latent Data

Areas of files and disks that are typically not apparent to the computer user (and often not to the operating system), but contain data nonetheless.

RAM Slack

The area beginning at the end of the logical file and terminating at the end of that sector. In some older operating systems this area is padded with information in RAM.

File Slack

The area that begins at the end of the last sector that contains logical data and terminates at the end of the cluster.

Unallocated Space

The area of the HDD that the operating system (file system table) sees as empty (containing no logical files) and ready for data. Simply stated, it is the unused portion of the HDD, but is not necessarily empty.

Courtesy of Peter Arnold, Inc.

Courtesy Getty Images, Inc.

Figure 17–1 Cutaway diagram of a personal computer showing the tangible hardware components of a computer system. *Courtesy Tim Downs*

Figure 17–2 An inside view of the platter and read/write head of a hard disk drive. *Courtesy Corbis RF*

Figure 17–3 Partitions of a hard disk drive

Figure 17–4 Rough sketch made at a crime scene with necessary measurements included.

Figure 17–5a Back of a computer showing all connections.

Figure 17–5b Back of a computer with each component correlated with its port through the use of a labeling scheme.

Figure 17–6 Screen shot of Encase Software. Encase is a common forensic sofware application capable of imaging and assisting in the analysis of data. *Courtesy of Encase, www.encase.com*

Figure 17–7 As user switches between applications and performs multiple tasks, data is swapped back and forth between RAM and the computer's hard drive. This area on the hard drive is referred to as either *swap space* or a *paging file.*

Figure 17–8 Slack space illustrated in a two-sector cluster. Cluster sizes are typically greater than two sectors, but two sectors are displayed here for simplicity.

Figure 17–9 File slack.

Figure 17–10 Simplistic view of a hard drive platter demonstrating the concept of unallocated space.

Andrew W. Donofrio is a Detective Sergeant with the Prosecutor's Office in Bergen County, New Jersey, and is a leading computer forensics examiner for Bergen County, with more than 18 years experience in the field of law enforcement. He has conducted more than 500 forensic examinations of computer evidence and frequently lectures on the subject throughout the state, as well as teaching multi-day courses on computer forensics and investigative topics at police academies and colleges in New Jersey. Det. Sgt. Donofrio writes regularly on Internet-related and computer forensics issues for a number of law enforcement publications and has appeared as a guest expert on Internet-related stories on MSNBC.

[1] A megabyte (MB) is approximately one million bytes (discussed later in the chapter), a gigabyte (GB) is approximately one billion bytes, or 1,000 megabytes.

[2] Pulling the plug should always be done by removing the plug from the back of the computer. If the plug is removed from the wall, a battery backup (UPS) might be in place, causing an alert to the system and keeping the unit powered on.

[3] In this instance, *bit* is both metaphorical and literal. Every bit of information is needed, so we must get it all. So too every bit, as in the smallest unit of data storage—a one or a zero—must be imaged.

[4] Actually, the more appropriate term is probably *paging* as opposed to *swapping*. This is because entire programs are typically not swapped in and out of memory to the swap space; rather, *pages* of memory are placed there.