What You Don't See On Your Hard Drive.
Brian Kuepper
April 4, 2002

SANS Security Essentials GSEC Practical Assignment Version 1.3

I. Introduction

Just because you don't see it doesn't mean it's not there. By having a knowledge of something that exists, but is hidden from your sight, will give you an advantage because you know it's there. In the security field it is very important to keep up to date on the latest information available. If you don't, someone will take advantage of your ignorance. Things are always changing and becoming bigger, better, faster and sometimes sneakier. A few years back in my Information Technology career I made the change from Desktop Support to the Information Security Group. Since then I have learned a tremendous amount about security. I have learned that you have to train yourself to think differently about things, add a little paranoia. This paper will address two security concerns that I found very interesting. They both have to do with things that are not in plain sight. The first security concern covers the issue of retrieving data that has been deleted. So many people have no idea about data that is left behind when you delete files or fdisk and format your hard drive. The second issue deals with hidden access and control of your computer. I will look at what a rootkit is and look at the recent development of rootkits designed for Microsoft Windows operating systems.

II. Historical Perspective

When Personal Computers first came out in the late 70's and early 80's they were, for the most part, considered toys. When IBM entered the Personal Computer market in 1982, the attitude towards Personal Computers changed. It seems like over night they transformed from toys to useful business tools. The Disk Operating System (DOS) that was originally installed on them was never really intended for commercial use. Early versions of the DOS operating system overlooked security as a part of its design. Compatibility issues with earlier versions of DOS and staying competitive in the market prevented addressing security issues in later upgrades to the operating system. Since all Microsoft Windows file systems today are derived from the DOS file system they all lack adequate security.

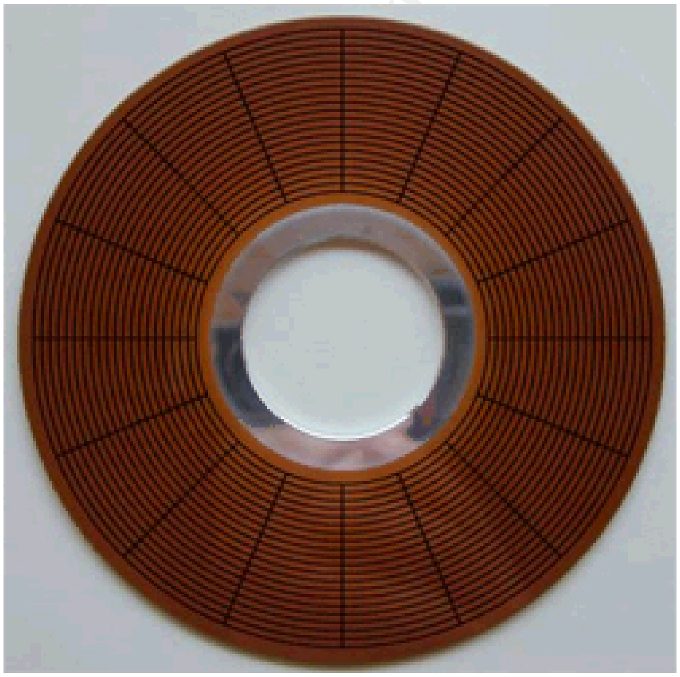
III. Data Storage

Data storage media like floppy diskettes, zip disks, and computer hard drives, are made up of materials that retain magnetic imprints of bit patterns. A bit is the smallest unit of measurement in the computer world. Bits are expressed in binary, the definition of Bi or Binary means 2. Binary is a base 2 numbering system that begins at zero (0) and ends with one (1). With the binary base numbering system there can only be two possible answers to a question, yes or no, black or white and so on. This works out perfectly for your computer, because the only thing it knows how to deal with is a ‘1’ or a ‘0’. In computer terms if a bit is turned ‘On’ it’s a ‘1’ and if it is turned ‘Off’ it’s a ‘0’.

A. Bytes

One byte is made up of eight bits. An example of a byte in binary is ‘01000001’. Computers use these eight bit numbers to perform calculations. For example, ‘01000001’ converted from binary to text represents the capital letter ‘A’. Different byte combinations represent numbers, letters, punctuation, etc. Bits and bytes make it possible for computers to work and store data. If you want to play around with text to binary conversion and vise versa, here is a web site that will do it for you: http://riot.com.au/bin/index.php3.

B. Sectors



[1]

[1] Kozierok, Charles M. “Tracks and Sectors.” Version: 2.2.0. April 17, 2001. URL: http://www.pcguide.com/ref/hdd/op/mediaTracks-c.html (April 4, 2002).

Bytes are stored on some type of magnetic media in blocks of data called sectors. Most computer hard drives and floppy drives use a sector size of 512 bytes. The sector is the smallest unit of storage that computers use to store data. Sectors are created and mapped when the computer storage device is low-level formatted. Low-level formats are usually performed at the factory by the manufacturer of the hard drive. The format pattern for floppy diskettes usually consists of binary data in the form of hex F6s which, converted to binary, is ‘11110110’. The same format pattern is sometimes used in the format of hard drives. The format patterns can consist of essentially any repeat character in every sector of the hard drive. During a low-level format, sectors are created and written consecutively to disk in concentric rings called tracks, as depicted in the picture above.

C. Clusters

All Microsoft operating systems read and write in blocks of data called clusters. A cluster is a fixed even number of sectors, e.g. 2048 bytes, 4096 bytes, etc. The number of sectors needed to make up a cluster is dependant on the type of storage device, the operating system, and the size of the logical storage device. Clusters are defined during the high-level format, which is performed by the operating system. New Technology File System (NTFS), FAT32 and FAT16 file systems will all have a different number of sectors per cluster. In my examples, I will be using Windows 2000 Professional formatted with NTFS and a standard cluster size of 4096 bytes (4 K).

D. Exception

The only exception to this format process is when floppy diskettes are involved. Floppy diskettes use the FAT12 file system, the low-level format and high-level format take place at the same time and they are performed by the operating system. If you were to check the box “Quick Format” under “Format Options” you are only performing a high-level format, which is why it’s quicker, essentially erasing only the File Allocation Table (FAT).

IV. File Handling

When you format a hard drive with the NTFS file system, the format process creates a Master File Table (MFT). Likewise the FAT16 or FAT32 (to accommodate disks larger than 2 gigabytes) file systems create a File Allocation Table (FAT). This has the same function as the MFT. Windows NT, 2000, and XP used in business environments generally use NTFS. Windows ME, 98, 95, 3.1 and DOS used primarily in home environments use the FAT16 or FAT32 file systems. To make things simpler and to avoid some confusion, I will concentrate on the NTFS file system. The FAT16 and FAT32 file systems have the same vulnerabilities.

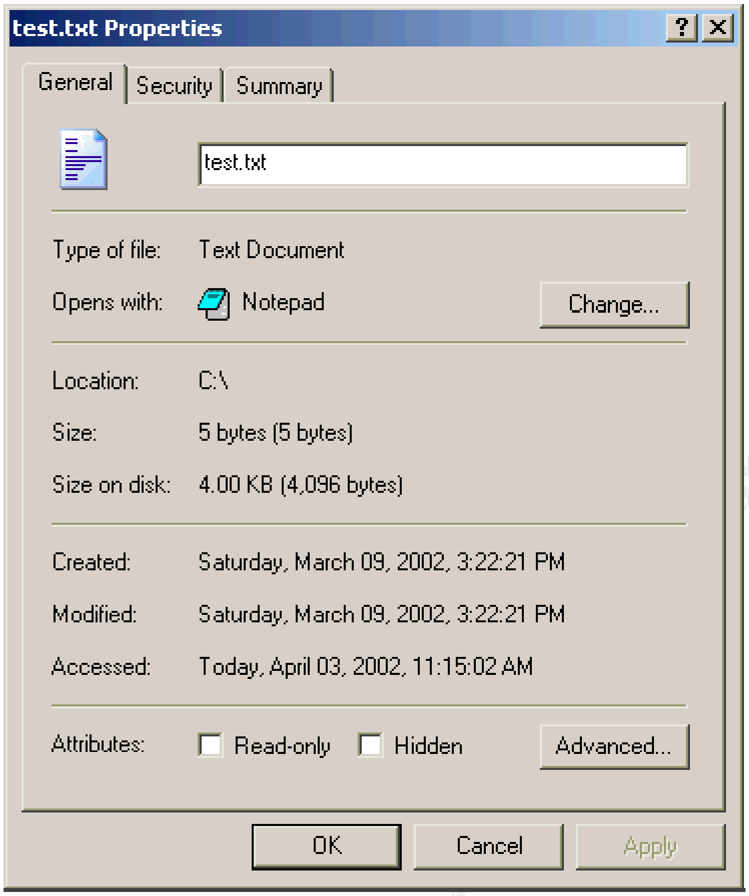
A. File Storage

When a file is created the NTFS file system can store the file in two ways. The first way addresses resident files, referring to a file small enough to be stored within the MFT record. The second way addresses non-resident files, referring to a file that is too large to be stored in the MFT record. Its disk location is stored in the MFT by one or more pointers to the data elsewhere on the disk. One thing that is important to know is only one file can be stored in a cluster at a time, so files can't share clusters. If a file is bigger than a single cluster then it spans several clusters. For example, on a hard drive with 512 byte sectors and 8 sectors per cluster, a 50 MB file would span 13 clusters and there would be an MFT entry made with data pointers referencing those 13 clusters.

B. File Deletion

Here is an idea to think about. When a file is deleted from the Recycle Bin in Windows or from the command prompt in DOS, that file is not really destroyed like you think it is. It's just hidden from your sight. Now lets backup a little and get up to date on some terminology. Free space is referred to as unallocated space and used space is referred to as allocated space on any type of data storage device, like your hard drive, for example. When you delete a file, the MFT record remains in the MFT, marked for deletion, until overwritten by a new MFT record. The data remains intact in the previously allocated clusters until overwritten by another file at a later date. Essentially what has happened is the file system has changed status of the cluster or clusters that the file was stored in from allocated space to unallocated space. The files remain in clusters on the hard drive waiting to be overwritten by another file in the future. Your free space, or unallocated space, on your hard drive contains deleted e-mail, documents, images, etc, that can be undeleted using a variety of free tools that can be download from the Internet. The main reason why data gets deleted in this fashion is because of performance reasons. Using space once occupied by a deleted file and overwriting it is faster than physically clearing the space after a file has been deleted. In the early development stages of files systems programmers were insistent on not losing data. It also results in better hardware life since one extra write operation is avoided every time a file is deleted, which, in theory, would double your hardware life.

V. File Slack



test.txt

Type of file: Text Document

Opens with: Notepad

Location: C:\

Size: 5 bytes (5 bytes)

Size on disk: 4.00 KB (4,096 bytes)

Created: Saturday, March 09, 2002, 3:22:21 PM

Modified: Saturday, March 09, 2002, 3:22:21 PM

Accessed: Today, April 03, 2002, 11:15:02 AM

Attributes: □ Read-only □ Hidden

Windows writes files to your hard drive in a minimum of 512-byte chunks to each sector. If you were to open Notepad and create a text file with the word “Hello” in it, it would be 5 bytes in size. As you can see in the graphic above, the file properties report that the file is using 4,096 bytes on the disk.

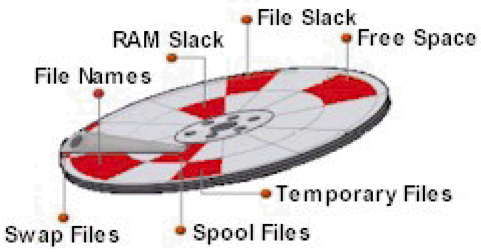
So what is in those 4,096 bytes on the disk? The actual information in the file, the word "Hello," is only taking up 5 bytes of space. This is where file slack comes into play. File slack is made up of two parts, RAM slack and disk slack. When the file system writes the file test.txt to the disk it has to write it in a minimum of 512-byte chunks. As you recall, our file was only 5 bytes so the file system has to fill up the remaining 507 bytes with something. It uses random information that it gets from the computer's RAM and sticks it at the end of the file to make up the full 512-byte size requirement. This could be any type of information - your password, credit card number, name, address, phone number, or any type of information that happened to be in your RAM at the time the data was written to your disk. Now that we have met the minimum requirement to write data to a sector, we need to meet the minimum requirement to write to a cluster, which is 4,096 bytes for my particular hard drive. The computer used in this test is a Windows 2000 Professional machine. The hard disk is set up with 512 bytes per sector, and 8 sectors per cluster. 512 bytes multiplied by 8 clusters equals 4096 bytes. This is where the file system allocates 4,096 bytes for the 5-byte file as seen in the example above. As you recall, when you delete a file, the MFT record remains in the MFT, marked for deletion, until overwritten by a new MFT record. In this specific example, the file is small enough to fit in the MFT record itself. If the file were larger, the MFT record would be marked for deletion and the data would remain intact in the unallocated clusters until overwritten by another file. Drive slack is the area of space between the end of a file and the end of the cluster holding the file illustrated in the example below.

```
Hello+++++++++++++++|----------------------------------------------------------------|(EOF)
| (5)  |     (507 Bytes)     |                          (3,584 Bytes)                          |
```

RAM Slack is indicated by "+"
Drive Slack is indicated by "-"
End of File (EOF)

The drive slack could be all or part of a deleted document or any type of data that was in the cluster previously. Drive slack could contain data from long ago, possibly from the first month you used your computer. Here is an analogy to explain drive slack better, say you record a two hour movie on a VHS tape with your VCR. Some time goes by and you decide that you don't care for that movie any more and instead of throwing the video tape away you use it to record the TV show Crime Scene Investigation (CSI) one night because you are busy that night. "CSI" is only an hour show, so the remaining hour of the movie that was on the video tape that didn't get recorded over by "CSI" is comparable to "Drive Slack". On large hard drives, it's possible to have as much as 700 megabytes of data.

A. What is in unallocated file space?



[2]

Unallocated file space on your hard drive potentially contains files that are

[2] "Understanding Basic File Drive/Data Terms."
URL: http://www.whitecanyon.com/library_understanding_terms.htm (April 4, 2002).

intact, remnants of files, directories, subdirectories, and temporary files, which were created and deleted by various computer applications and the operating system. I downloaded some un-deletion software from the Internet and I was amazed at what I found on my hard drive hiding in plain sight. The following examples describe how data and information can end up in unallocated file space.

1. Spool Files

   By default when you install your printer it is setup to spool your print jobs to a spool file located on your hard drive. Your computer is much faster at processing documents to be printed than your printer is, so the print spool queues them up and the printer prints them as fast as it can. When you print the contents of a word processing file, Windows writes a copy of the file to a temporary file on the hard disk while waiting to get the attention of the printer. Once the file has been printed, Windows "deletes" the temporary file. The data in your print job is now stored in unallocated storage space on the hard drive.

2. Temporary Files

   Many computer applications create .TMP (temporary) files on your computer while they are running. These .TMP files are needed while the program is running to keep track of changes made to the original file that was opened. A good example of this is with Microsoft Word. When you open a document in Word it creates a temporary copy of the file that you're working on and records all the changes that are made. When you are finished it saves all the recorded changes from the .TMP file into the original file, updating it. The .TMP file is then deleted and becomes a part of your hard drive's unallocated space.

3. Email Files

   Email has become a very effective communication tool. It plays a big part of our everyday business and our personal lives. Mail is private, whether it is electronic or paper. Email messages are stored on your hard drive just like all of your other files. When deleted it is not removed. It also becomes a part of your hard drive's unallocated space, with the potential of being read by anyone.

4. Temporary Internet Files

   Internet browsing is responsible for large amounts of unallocated space. Every time you visit a web page, the contents of that page are downloaded to your "Temporary Internet Files" directory. The size of your Temporary Internet Files directory in Internet Explorer usually is equal to your computer's RAM. So when that mischievous user visits some questionable sites and then deletes the temporary Internet files, thinking all evidence has been removed, they are mistaken.

5. Installation Files

When you install programs on you computer, during the installation process all sorts of temporary files are copied to your hard drive and then erased at the end of the installation process.

6. Peer-to-Peer Files

With the popular use of Peer-to-Peer ("P2P") file sharing programs like Morpheus, KaZaA, Gnucleus, Bear Share, and Lime Wire, just to name a few, people download all types of files with high speed Internet connections in just about everyone's homes these days. Hundreds of megabytes are being copied to hard drives and then later being deleted.

7. Swap Files

Windows swap files can be temporary or permanent, depending on which Microsoft operating system you have. Windows 95, 98, and ME are set up to have the swap file expand and contract depending on how much it space needs. To be able to run multiple programs at the same time with only so much RAM, Windows uses this swap file to expand the computer's RAM. Windows swaps an application's data to and from the swap file as needed, based on what is actively running. Because this file contains application data, there could be almost any type of data in it. Things like passwords, complete or partial documents, email, credit card numbers, images, or anything that could be in memory. Windows NT, 2000, and XP use permanent swap files that don't expand and contract. By default Windows 2000 sets the swap size to the amount of RAM * 1.5. As a side note, there is a local security policy setting under Start, Programs, Administrative Tools, Local Security Policies, Local Policies, Security Options that allows you to set Windows to 'Clear virtual memory page when Windows shuts down'.

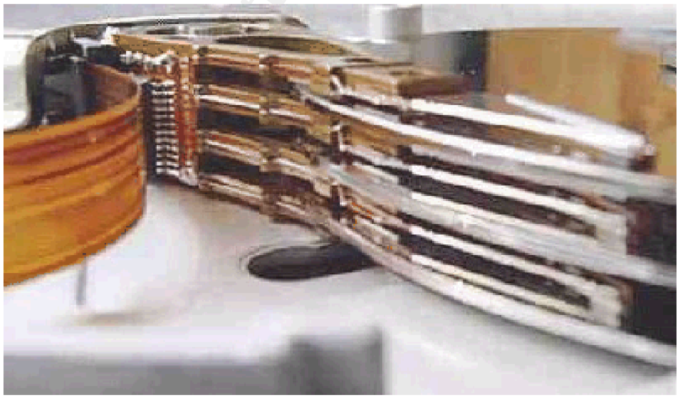
8. Partial Files

Have you ever copied a file to a zip disk or floppy disk? Then three quarters of the way through an unsuspected error message pops up, “There is not enough free disk space. Delete one or more files to free disk space, and then try again.” A review of the directory where you were copying the file shows no entry for the file that was partially written. What has happened is Windows created a pointer record in the File Allocation Table (FAT) and wrote as much data to the disk as it could. When it discovered that there was no more available unallocated space left to write the remainder of the file, it marked the record for deletion, and left the partially written data on the disk.

9. System Crash

Another way partial files could end up on any type of storage media is if the operating system or an application was writing data or had temporary files open and there was a power failure or system crash. All of those files were unable to be updated in the computer's Master File Table. Now they are sitting out there in unallocated space.

B. Shadow Data



[3]

It doesn’t matter what kind of storage medium you use to store your data;

[3] Brain, Marshall. “Inside a Hard Disk.”
URL: http://www.howstuffworks.com/hard-disk2.htm (April 4, 2002).

floppy diskette, zip disk, hard disk, or tape. These storage mediums all have one thing in common; they use a mechanical head to electronically write binary data in magnetic patterns of ones and zeros. As mentioned previously, sectors are written in consecutive rings called tracks, the mechanical heads that write the binary data to each platter use magnetically recorded servo-patterns to allow control and movement of the read/write head assembly. This process of writing data happens with less than exact tolerance levels. It's pretty much impossible for the head to be exactly centered over the track. Horizontal and vertical head alignment is just a bit different every time data is written and rewritten to the same track. The fringe data that remains on the very edges of the track is called Shadow Data.

Shadow data contains remnants of data that was written previously to a track, located slightly outside the track's last write path.

There are two phenomena which cause shadow data to come about: mechanical and magnetic issues. Maybe this analogy will explain it better. Imagine a long straight road with a white 4-inch stripe down the exact center of the road. The highway department has been told to go out and repaint over that white 4-inch stripe with a yellow 4-inch stripe. So they go out and do this. If we were to go out and examine the line you would notice that the yellow paint wouldn't cover all the white paint perfectly. As the truck with the paint sprayer drives down the road, due to mechanical imperfections the truck weaves back and forth in an elongated "s" pattern. This is the same thing that the heads in a hard drive do, when they write the data; the head wobbles horizontally in an elongated "s" pattern due to imperfections in the drive's servo mechanism which controls the arm that the heads are attached to. That covers the mechanical issue.

Now let's look at the magnetic issue. As the spray paint truck drives down the road, the paint head is also going to move up and down to create a vertical elongated "s" pattern as the line is painted. Greater distance between the road and the paint head creates a slightly wider line. Likewise, the closer the paint head is to the road the narrower the line will be, which would not completely cover the line. This applies to the heads in your hard drive. This over spray effect is actually caused by variances in signal strength as the head moves up and down vertically while the data is written to the track.

Now that we have covered horizontal shadow data, let's look at vertical shadow data. The material used to store the magnetic imprint consists of iron oxide layers that have been sprayed on the platter of a hard disk drive or the Mylar sheets used in floppy diskettes. Keeping with the spray paint analogy, if you were to get down on your hands and knees and look closely at the asphalt road, you would notice that the pavement is porous.

When the paint is sprayed on to the road, some of it will seep down into the pores of the pavement. If you were to slice a thin layer off the top of the asphalt, the original painted line will be overlapped with the new color in some places, but remnants of the original paint would also exist. Similarly, on hard disk platters and floppy Mylar sheets, layering occurs due to the physical flaws in the storage media and variances in the ability of the iron oxide to hold a magnetic charge.

Newer hard dive technologies make recovery of shadow data a bit less fruitful, because the drives densities require higher frequencies to write data to narrower tracks. Some of the older hard drives with slower RPM's and wider tracks produce more shadow data. For beginners interested in experimenting with shadow data, floppy diskettes might be a good place to start, because the technology is so old and the RPM's are very slow, hence more wobble. From a forensics standpoint, shadow data has yet to become a reliable source of computer evidence, mostly because of the cost involved. You have to remove the platters from the hard drive housing and process them using very expensive and specialized equipment. This cost factor usually protects the private sector computer users and corporations. However you should still be aware that someone may be willing to pay the huge price to steal corporate and government secrets stored in the form of shadow data. Microsoft spends hundreds of thousands of dollars every year on data security. I bet that you don't see them throwing old hard drives in the dumpster. Different ways to protect yourself against shadow data will be discussed in an upcoming section titled, "Data Security."

II. Erasing Data

I used to think that when you ran the utilities Fdisk and Format on a hard drive it erased everything. That is not the case; the only thing that is erased is the file structure. The drive can not be recovered back to its original state, but the data stored on the drive can still be read. When you format an IDE hard drive using the command "format c:" from a MS-DOS boot disk, you are doing what's called a high-level format, which creates a directory and initializes the drive. High-level formats don't erase every sector on the disk, just the file table. It kind of makes you think twice about getting rid of that old computer. In doing my research, I stumbled across a newspaper article entitled, "Donor doesn't want PC to keep on giving." [4]

A. Low-Level Format

There are rumors running around the internet and with your computer buddies that you can do a low-level format on an IDE or ATA hard drive.

[4] Marshall, Patrick. "Donor doesn't want PC to keep on giving." Seattle Times. July 23, 2000. URL: http://seattletimes.nwsource.com/ptech/html98/mrsh23_20000723.html (April 4, 2002).

This is not true. If you were to perform a true "low-level format" it would destroy the drive or at least slow it down radically. Writing sectors to hard drives can only happen at the factory where the drive is manufactured. Most hard drive manufacturers have a special utility available for download (they call it a Low-Level Format utility) that will erase the data in every sector of you hard drive. These are manufacturer specific, so don't try a Maxtor utility in a Quantum drive.

B. Wipe Utilities

There are several file wipe utilities that run within Windows and are available on the Internet. I downloaded one and tried it on my hard drive. It claimed to wipe all the unallocated space on my disk, leaving the allocated space alone. It also claimed to write over the file slack on my drive. I scheduled it to run during the night. In the morning I loaded the trial version of some undelete software that I had downloaded from the Internet. I was able to see and undelete a large amount of files from free space that was supposed to have been wiped clean. Time constraints prevented me from trying other brands of software. There may be software out there that does this and that could be the topic of a paper on its own. I was not impressed.

Windows XP contains a "wiping" utility in the Professional Edition. There are several articles on the Internet with mixed reviews on this. An article at pcworld.com entitled, "Former Fed Says XP Poses a Security Threat," said:

"Michael Anderson, president of New Technologies, says data scrubbing features in Windows XP Professional will make it impossible for federal agents and law enforcement to find and reconstruct digital evidence buried on computers, particularly those seized from terrorists." [5]

The utility is a command-line program that provides an alternate method for managing the Encrypting File System ("EFS") and also has a "wipe" feature that overwrites data in unallocated clusters. The program makes three passes of writes over the unallocated space on your hard drive. The first pass is hex 00, the second hex FF and the last pass is random characters, making the data in those unallocated clusters un-recoverable. This utility would appear to comply with the Department of Defense 5220.22-M disk- sanitizing standard, which states:

"'Non-Removable Rigid Disks' or hard drives must be sanitized for reuse by 'Overwriting all addressable locations with a character, its

[5] Fontana, John. "Former Fed Says XP Poses a Security Threat." October 15, 2001. URL: http://www.pcworld.com/news/article/0,aid,66023,00.asp (April 4, 2002).

> *complement, then a random character and verify.'"* [6]

Here is an example of the program output:

```
To remove as much data as possible, please close all other applications
while running CIPHER.
Writing 0x00
...................................................................................
Writing 0xFF
...................................................................................
Writing Random Numbers
...................................................................................
```

> Guidance Software ran some tests with the Windows XP Professional "wiping" utility. Cipher.exe was used to wipe all unallocated clusters from the root folder. Guidance reported that all unallocated space was filled with random values, and a very small portion of the MFT. Only 10 to 15 records were overwritten in the MFT, and the majority of the records marked for deletion went untouched. They also observed that the utility didn't affect file slack, registry files, the paging file (swap file), or file shortcuts. Overall, the command-line interface and lack of a scheduling feature is unpopular. Also I could find very little documentation supporting this utility.
>
> In my opinion, the Windows XP "wiping" utility and its overall effectiveness seems to be about par for the course in overwriting 100% of unallocated file space and records marked for deletion in the MFT. I don't see why everyone is "up in arms" over this. There are several third-party wiping utilities available for download on the Internet that do just about the same thing. I realize that the people that are complaining are in the computer forensics field, and these tools probably make their job harder. I think we have a right to be secure - it's our data. I haven't heard of anyone complaining about people shredding paper documents.

III. Data Security

One great way to protect your data from being read is to use encryption. There are several companies that offer software that encrypts your entire hard drive. PC Guardian ® automatically decrypts individual files before they are loaded into memory and when data is written back to the hard disk, it is automatically re-encrypted. The secure feature about this is only individual files are decrypted at any one time, not the whole hard disk. The drive itself is password protected. On

[6] Stone, Kimberly and Keightley, Richard. "Can Computer Investigations Survive Windows XP?" December 2001.
URL: http://www.guidancesoftware.com/html/XPwhitepaper.pdf (April 4, 2002).

boot up, if you don't enter the correct password your hard drive stays encrypted and your information is safe. After a 3 failed attempts, the encryption software locks you out from any further attempts to guess the password and the computer must be rebooted. This works out great for people with laptops that travel who worry about theft and loss of confidential data.

The U.S. Government treats the protection of Confidential or Top Secret digital data as high priority. Some methods used by the Naval Information Systems Management Center to destroy magnetic data storage media include incineration, degaussing, sanding off the oxide disk surface with a disk sander, or destruction of oxide disk surfaces with hydriodic acid. As you can see, they go to great extents to make sure that it will be physically impossible to read any discarded data.

For most people in the private sector booting from a modified DOS boot disk and using a wipe utility like Byte Back or booting from a Linuxcare Credit Card CD and using *"dd if=/dev/zero of=/dev/had bs=512."*[7] will wipe a hard drive clean. These are methods that people in the computer forensic field use to prepare their "evidence" drives. These utilities write a repeating character in every sector. There is an excellent paper that addresses data security by Peter Gutmann entitled *"Secure Deletion of Data from Magnetic and Solid-State Memory."*[8] You can find the web page by going to the reference in the footnotes. I saw several wiping utilities on the web while doing my research that incorporate this wiping algorithm into their software. Keep in mind that overwriting a hard drive 35 times with different data is going to take a toll on the life of your hardware. If you're really paranoid, you can disassemble your hard drive, removing the magnets and platters, destroying the magnetic servo patterns and the disk order, making the hard drive useless. The only way to recover the data under those circumstances is to pay some very serious money to a forensic company to use specialized tools to extract the data.

IV. Hidden Access to Your Data

We have been talking about how others can extract hidden data from your hard drive once discarded. Now we are going to shift gears and talk about how others can access your data remotely and hide data on you hard drive without your knowledge. The popularity of the Internet has created a wonderful place for hackers to play around. It seems like every time you turn on the news or read your email you hear about someone getting hacked. I know that I was quite surprised at the amount of people trying to get into my dad's computer. He has a broadband connection to the Internet so I installed Black Ice (personal firewall

[7] Holley, James. "September 2000 Market Survey Computer Forensics." September 2000. URL: http://www.scmagazine.com/scmagazine/2000_09/survey/products_01.html (April 4, 2002).

[8] Gutmann, Peter. "Secure Deletion of Data from Magnetic and Solid-State Memory." July 1996. URL: http://www.usenix.org/publications/library/proceedings/sec96/full_papers/gutmann/ (April 4, 2002).

software) on his system and someone is always scanning his system for vulnerabilities that they can exploit. In the event someone is successful at breaking into your system, whether it's a personal computer or a corporation's server with social security and credit card numbers, it is something we need to be aware of.

A. Trojan Horses

I'm sure that we are all familiar with the story of the Trojan horse, and the city of Troy. A Trojan or Trojan horse in computer terms is a piece of malicious software that hides inside an otherwise innocuous piece of software. For example, you download what looks to be a "free" fun game to play like *"Whackamole."* You start the game and begin to play. Meanwhile, 'NetBus' is being installed on your PC without your knowledge and any person on the Internet with NetBus can snoop around on your computer, look for important files, passwords, and your personal information. As a side note, there really is a Trojan version of "Whackamole" out there on the Internet so be careful what you download. Most anti-virus software scans files for Trojans and will alert you when one is found. In doing my research for this paper I visited some "underground" hacker web sites and downloaded some zip files only to find Trojan versions of the utilities I was looking for, Luckily, Norton Anti-Virus saved me. I also run Zone Alarm on the same PC and watch for Trojans that Norton might have missed. While playing around with KaZaA, I have noticed that you can download a variety of software that people are sharing that could have been modified and have Trojans nested in them. You have to be really careful these days. There are all kinds of Trojan programs out there on the Internet. They have been around for quite some time now and are growing in popularity. To avoid being infected with a Trojan, download and execute files only from trusted sources.

B. Backdoors

A backdoor is an unauthorized way of gaining access to a program or computer system. In the example above I referenced NetBus. This program is essentially a backdoor type program. It allows the hacker to: open/close CD-ROM, swap mouse buttons, start optional applications, play a .wav file, control the mouse, popup messages on your screen, shut down Windows, download/upload/delete files, go to an optional URL, send keystrokes and disable keys, turn the sound-volume up and down, and record sounds from the microphone, just to name a few of its features. If you are reading this and it sounds familiar, you may want to check out your system. A hacker can get in through a backdoor installed on your system as long as it is running and has not been discovered and disabled. Typically, hackers have to look for vulnerabilities in a system that lets them in to a computer. With a backdoor program there is no more "rattling door knobs" to find one open, now they have a key.

V. What is a Rootkit?

The definition of a rootkit is a collection of tools that hide the presence of the attacker while giving the hacker the ability to keep full control of the system continuously without being detected. All rootkits can be grouped into two types; application rootkits, and kernel rootkits. An application rootkit is where the hacker replaces a good system application with a trojaned system file. The trojaned system file will provide the backdoor, hide the hacker's presence and not log any connection and activity by the hacker. A kernel rootkit is very powerful and less detectible than an application rootkit because it exploits and manipulates the kernel, bypassing conventional system integrity checks. Because of this they are very hard to detect. An attacker will scan the Internet with a utility looking for a system that has a vulnerability they know how to exploit. Once the attacker gets access to the system through the known vulnerability, they plant a rootkit on the system and activate it, giving them root access to the system through some type of backdoor. Sometimes the attacker will patch the vulnerability on the system to keep other attackers out. The rootkit is hidden and undetectable to most users. Rootkits, formerly a Unix concept, are spreading to the Windows NT/2000 Operating Systems. Greg Hoglund and a group of developers have developed a rootkit for Windows NT/2000 called "NT rootkit".

A. NT Rootkit

In doing my research I came across a web site with some very interesting information at www.rootkit.com. Last time I checked the web site was down. I was unable to download the zip file containing the NT rootkit from the web site. The only thing that was available was the root_readme.txt, which discussed its features. It sounded pretty interesting, nothing I like I have ever seen before. The hunt was on. I had to find the actual rootkit so I could play with it. After a couple of weeks searching the web I finally found _root_040.zip. Its functionality is amazing. Here are the key features found in version 0.40 and features being developed for future releases as explained in the root_readme.txt file:

- Embedded TCP/IP Stack (stateless)
- Keyboard Sniffing (to be added in the future)
- Hide Processes
- Hide Files and Directories
- Hide Registry Entries
- Command Shell (to be added in the future)

- EXE Redirection
- Cause a blue screen of death (BSOD)
- See anything hidden while prefix-hiding is on

Here are my findings after playing around with the rootkit for several days. To install the rootkit, extract out deploy.exe and _root_.sys from the zip file to a common directory. Double click deploy.exe. A DOS window pops up and then disappears. If you refresh the view of the Explorer window, you will notice that _root_.sys has disappeared. To stop and start the rootkit in real-time you can use the following commands: net stop _root_ and net start _root_, respectively.

1. **Embedded TCP/IP Stack (stateless)**

   The stateless TCP/IP stack works by determining the state of the connection based on the information contained in the incoming packet. The NT rootkit also has a hard coded IP address of 10.0.0.166 so that any machine with a 10.x IP address should be able to telnet to the rootkit on any port, and more than one person can login to the rootkit at once. The only drawback is, if two people issue commands at the exact same time it can cause problems. A little side bonus, because NT rootkit uses its own stack it doesn't show up in netstat.exe. One thing you will want to look out for is an IP address conflict; you should make sure that there isn't a machine on your network with that IP address. Otherwise, the two computers will get into an ARP war. The rootkit uses raw connections to your Ethernet card, giving developers the option to add more functionality to the rootkit. I'm sure in later versions packet sniffing could be a new feature along with many other things.

2. **Keyboard Sniffing**

   The version that I downloaded had keyboard sniffing disabled. You can comment the line DriverEntry() back in to enable it. NT rootkit hooks into the keyboard driver on the machine and redirects all key strokes to go through itself, intercepting all key strokes typed at the console. This enables the hacker to eavesdrop on the system console. This is done by issuing the 'sniffkeys' command from the k-mode shell. It looks something like this if you issued the sniffkeys command at the k-mode shell and waited until the administrator logs in by typing his user ID and password, and then opens a command prompt and issues a dir command:

```
¿sniffkeys
```

```
keyboard sniffing now ON
----------------------------------------
--administrator--h@ppyd@y--dir--
```

3. **Hide Processes**

NT rootkit also has the ability to hide processes from view in the Task Manager’s process list or the command line Resource Kit utility pulist.exe. All you have to do is rename the program to begin with the characters _root_. The rootkit filters those programs to hide them from view.

4. **Hide Files and Directories**

Hiding files and directories works pretty much the same way as hiding processes. All you have to do is rename the file or directory with the characters _root. With the rootkit installed and running, create a directory in the root of C:\ and call it Rootkit, then put a copy of CMD.exe in the rootkit directory, then rename CMD.exe to _root_CMD.exe, you will notice that after you strike the Enter key the renamed file will disappear from site. It’s still there. You just can’t see it. Rename the directory Rootkit to _root_Rootkit, likewise it disappears. The rootkit works by patching the NT kernel and intercepting the system call to list files and directories, all files and directories that begin with the sequence _root_ are hidden from display. (It hides processes with this same methodology.) The only flaw in this scenario is that files and directories are only hidden from the compromised machine. If you map a drive remotely from another machine to the administrative share (\\compromisedmachine\c$) on the compromised machine you can see the hidden files. This is because the remote machine is using a clean kernel to view the files and directories on the compromised machine, avoiding the rootkits filtration process.

5. **Hide Registry Entries**

You can add or rename any value name or key name in the registry to begin with the characters _root_ and they will be hidden from view. This works with regedit.exe and regedt32.exe, because the rootkit once again intercepts the system call for the listings in the registry and hides them. I manually created some keys and values and renamed them to begin with the characters _root_. I searched the registry with Regedit.exe and the command line utility regfind.exe found in the Windows Resource Kit, and didn’t find the hidden keys and values I created. The only flaw I could find in the registry when I did a search was the unhidden location of the _root_ service, which would tell you if this version of the rootkit had been installed on your machine. I exported the registry keys where the service is located:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Root\LEGACY__
ROOT_]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Root\LEGACY__
ROOT_\0000]
"Service"="_root_"
"Legacy"=dword:00000001
"ConfigFlags"=dword:00000000
"Class"="LegacyDriver"
"ClassGUID"="{8ECC055D-047F-11D1-A537-0000F8753ED1}"
"DeviceDesc"="_root_"
"Capabilities"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Root\LEGACY__
ROOT_\0000\Control]
"DeviceReference"=dword:fd938630
"ActiveService"="_root_"
```

As you can see, the reason that the rootkit service was not hidden is because the Key Name is LEGACY__ROOT_ and doesn't begin with _root_. I'm sure that someone will figure out a way around this. "Where there's a will there's a way" and the rootkit is only in Beta form right now.

6. **Command Shell**

This is code that connects to a specific TCP port listening on the compromised machine and binds a command shell ("CMD.EXE") to it. Quoted from the root_readme.txt in the rootkit:

"We have experimented with launching win32 processes from kernel mode. This has been non-trivial. We have demonstrated this working at Blackhat - but the feature is disabled in this build. It will be added back in for the 044 branch - but there are many kinks still being worked out."[9]

[9] Hoglund, Greg. "NT Rootkit."

The added command shell will really make this tool dangerous.

7. EXE Redirection

Under the heading Test EXE redirection in the root_readme.txt it reads:

"For now, this test is hard coded." [10]

This leads me to believe that once the bugs get worked out it will not be hard-coded and left wide open to many possibilities. To test EXE redirection, I copied calc.exe to C:\ and then copied CMD.EXE to C:\ and renamed it to _root_cmd.exe. (For this hard-coded test to work there can't be any other exe's in the root of C:\, just calc.exe and _root_cmd.exe.) I clicked 'Start', 'Run' and typed '_root_cmd.exe' and clicked 'OK'. The Calculator popped up on the screen. The rootkit detected the execution of a file name that started with '_root_' and redirected it to "C:\calc.exe". With the rootkit only becoming involved when the file execution passes through the Kernel, it should be able to fool programs that perform CRC's or Hashes of files looking for Trojans. I am sure that they put the hard-coded version of this feature in the rootkit to demonstrate functionality and that it works, until they can fine tune the non-hard-coded version.

8. Blue Screen of Death (BSOD)

NT rootkit has the functionality to issue an interrupt, causing a blue screen of death. All you have to do is telnet to the hard-coded IP address 10.0.0.166, which starts the k-mode shell, and issue the command 'debugint' and - whamo - the computer blue screens. What a sense of power.

9. Reverse Hiding

This is a pretty neat feature. If you copy regedit.exe to C:\ and rename it to _root_regedit.exe, it will hide itself after the rename. Launch it from the 'Run' command and you will notice that you can see all the hidden _root_ entries. This works with any program; _root_cmd.exe can see the hidden files and directories when you issue a 'dir' command. _root_taskmgr.exe can see hidden process running. I think you get the idea. I think that's a pretty cool feature.

URL: http://www.megasecurity.org/Tools/Nt_rootkit_all.html (April 4, 2002).

[10] Hoglund, Greg. "NT Rootkit."
URL: http://www.megasecurity.org/Tools/Nt_rootkit_all.html (April 4, 2002).

10. Beware

The last thing that you should know about the NT rootkit is _root_.sys is a driver and it does show up when you run the utility drivers.exe from the Windows Resource Kit. The entry is listed when drivers.exe is executed; all the other drivers have detailed information about them except for _root_.sys, which has no detailed information at all, just its name. Windows rootkits are currently not as widespread as Unix rootkits. Remember, because the NT Rootkit is in development we rely on hard-coded strings such as _root_ for detection. The possibility exists to hide malicious code in existing kernel-mode drivers (ex: NULL.SYS) making it far more stealthy and difficult to find. As the NT Rootkit becomes more prevalent, you can expect that other attackers will modify the strings, creating different variations of the NT Rootkit.

VI. Conclusion

I hope that I have been able to shed some light on vulnerabilities that you were unaware of so that in the future you will take proper measures to keep your data secure. File deletion is a subject that most people don't know about and I think that it needs more public attention. With all these file-wiping vendors advertising their products on shareware sites, hopefully awareness will increase. If you try out any of these products, don't take their word that all the files are erased. Check it out for yourself. If your computer is connected to the Internet, particularly if you are "always on" as in Cable or DSL connections, take precautions that are necessary to keep your data safe from attackers. Options include hardware solutions like a Broadband/DSL firewall/4-port switch from Linksys, or software firewall solutions like Zone Alarm, Black Ice, and Norton Anti-virus 2002, just to name a few. Having an unprotected system is just asking for trouble.

References:

Alcaraz, Juan. "CS-384 Operating Systems section 001Term Paper File Management in operating systems." January 29, 1999.
URL: http://people.msoe.edu/~barnicks/courses/cs384/papers9899/alcaraz.pdf (April 4, 2002.).

Brain, Marshall. "Inside a Hard Disk."
URL: http://www.howstuffworks.com/hard-disk2.htm (April 4, 2002).

Bryson, Curt and Anderson, Michael R. "Shadow Data The Fifth Dimension of Data Security Risk." August 15, 2001.
URL: http://www.forensics-intl.com/art15.html (April 4, 2002).

Fontana, John. “Former Fed Says XP Poses a Security Threat.” October 15, 2001.
URL: http://www.pcworld.com/news/article/0,aid,66023,00.asp (April 4, 2002).

Gutmann, Peter. “Secure Deletion of Data from Magnetic and Solid-State Memory.” July 1996.
URL:
http://www.usenix.org/publications/library/proceedings/sec96/full_papers/gutmann/ (April 4, 2002).

Hoglund, Greg. “NT Rootkit.”
URL: http://www.megasecurity.org/Tools/Nt_rootkit_all.html (April 4, 2002).

Holley, James. “September 2000 Market Survey Computer Forensics.” September 2000.
URL:
http://www.scmagazine.com/scmagazine/2000_09/survey/products_01.html
(April 4, 2002).

Kozierok, Charles M. “Master File Table (MFT).” Version: 2.2.0. April 17, 2001.
URL: http://www.pcguide.com/ref/hdd/file/ntfs/archMFT-c.html (April 4, 2002).

Kozierok, Charles M. “Tracks and Sectors.” Version: 2.2.0. April 17, 2001.
URL: http://www.pcguide.com/ref/hdd/op/mediaTracks-c.html (April 4, 2002).

Manap, Saliman. “Rootkit: Attacker undercover tools.”
URL: http://www.niser.org.my/resources/rootkit.pdf (April 4, 2002).

Marshall, Patrick. “Donor doesn't want PC to keep on giving.” Seattle Times. July 23, 2000.
URL: http://seattletimes.nwsource.com/ptech/html98/mrsh23_20000723.html
(April 4, 2002).

Prosise, Chris and Shah, Saumil Udayan. “Hackers' rootkit for NT.” Security Issues.
February 22, 2001.
URL: http://webbuilder.netscape.com/webbuilding/0-7532-8-4877567-1.html
(April 4, 2002).

Prosise, Chris and Shah, Saumil Udayan. “Stop Windows hackers.” Security Issues.
March 8, 2001.
URL: http://webbuilder.netscape.com/webbuilding/0-7532-8-4996985-1.html?tag=st.bl.7532.edt.7532-8-4996985-1 (April 4, 2002).

Stone, Kimberly and Keightley, Richard. "Can Computer Investigations Survive Windows XP?" December 2001.
URL: http://www.guidancesoftware.com/html/XPwhitepaper.pdf (April 4, 2002).

"Cluster Defined." December 28, 2001.
URL: http://www.forensics-intl.com/def19.html (April 4, 2002).

"Encryption Plus® Hard Disk 6.1."
URL: http://www.pcguardian.com/software/hard_disk.html (April 4, 2002).

"File Slack Defined." October 4, 2000.
URL: http://www.forensics-intl.com/def6.html (April 4, 2002).

"How Secure Are You?"
URL: http://www.whitecanyon.com/library_how_secure.htm (April 4, 2002).

"Information Systems Security (INFOSEC) Program Guidelines." Navy Staff Office Publication. NAVSO P-5239-26. September 1993.
URL: http://www.fas.org/irp/doddir/navy/5239_26.htm (April 4, 2002).

"Low Level Formatting an IDE Hard Drive."
URL: http://freepctech.com/pc/001/007.shtml (April 4, 2002).

"Riot.com.au text to binary converter"
URL: http://riot.com.au/bin/index.php3 (April 4, 2002).

"Sector Described." December 28, 2001.
URL: http://www.forensics-intl.com/def15.html (April 4, 2002).

"SecureClean."
URL: http://www.accessdata.com/Product05_LearnMore.htm (April 4, 2002).

"Unallocated File Space Defined." October 24, 2000.
URL: http://www.forensics-intl.com/def8.html (April 4, 2002).

"Understanding Basic File Drive/Data Terms."
URL: http://www.whitecanyon.com/library_understanding_terms.htm
(April 4, 2002).

"Windows Swap File Defined." October 4, 2000.
URL: http://www.forensics-intl.com/def7.html (April 4, 2002).