Global Information Assurance Certification Paper

Is it really gone ? (A look at data deletion)

Introduction

What happens to data once it's been deleted? Is it safe to assume that it's really gone, never to be seen again?

Unfortunately this is what many people think, they assume that once they delete a file the Operating System they are using actually removes the file from the storage media. Thus making it impossible to retrieve.

This is not the case, as we will see later on in this paper.

The goal of this paper is to create an awareness surrounding the secure deletion of data. This paper will look at the various ways of deleting data and will attempt to highlight the shortcomings of each of these methods.

Delete Commands

The quickest and easiest way to "erase" data is to use the Operating Systems (OS) built in delete commands, these commands may vary from OS to OS but they all do pretty much the same thing.

When the delete command is used it doesn't actually touch the data recorded on the media. It only removes the index entry and pointers to the actual data so that it appears as if the file has been removed. The space that was allocated to the data is then made available to the OS for future write commands.

This process is very insecure and only offers protection from general computer uses snooping around. There are many utilities available which would allow any knowledgeable user to move beyond the operating systems file indexing structure and examine the data stored on the disk directly, thus giving access to previously deleted data.

Most OS's even provide access to the raw disk data with commands such as undelete (DOS, Windows), grep, dd and debugfs (UNIX).

There is a way to prevent a deleted file from being reread by such utilities. This can be done by overwriting the sectors used by the file with a new data pattern, then if the file is recovered the information contained will be useless. This process is an improvement but still posses some security risks.

Another problem arises in the way the operating system and applications function. Most operating systems applications create temporary files or swap files while they are working with the data. When the applications are closed or finished with the data they "erase" the temporary files. This practice provides another source of data to be searched for with undelete utilities. So even if the original file has been overwritten, multiple copies of the raw data may still exist in various unused parts of the disk drive.

Re-Formatting or Re-Initialising

There are two major different types of formatting; these are low level formats and Operating System format commands.
Low-level formats are generally the better of the two as they re initialise the disk by writing either 1's or 0's to the media as apposed to an Operating Systems format which normally only create a new indexing scheme for the Operating System, making all the sectors available for the writing of new data and leaving the old data intact.

It is a good idea to familiarize one self with the exact actions of the format commands used by your organizations Operating Systems. As there are many different ways to issue the format commands.

Under DOS based systems you have the option to run the format command with the /U parameter (e.g. format c: /U), this will prevent the format process from being reversed by the unformat command.

The same is true for Windows, which has the options of a quick format and a complete format. With the complete format being the far safer option between the two.

Overwriting of the Data

A slightly greater level of protection for erased data can be attained by overwriting the entire media or the sectors used by the erased data. To do this we need to overwrite the data area as many times as possible with alternating patterns. In selecting the data pattern to write to the disk, the aim is to try and to switch each magnetic domain on the disk back and forth as much as possible (the same concept as degaussing) without writing the same pattern twice in a row.

A few factor must be taken into account that complicate the choice of the pattern to be written, these include such things as:

> The frequency at which the drive writes data to the media this is because very high frequency signals only scratch the surface of the media. A low frequency signal is required to penetrate the media as much as possible.
>
> The way that disks use a form of run-length limited (RLL) encoding, so that adjacent 1's are not written. This encoding is used to prevent the drive from losing track of where it is in the data, by making sure that transitions aren't placed too closely together or too far apart.

The following table shows a sequence of 35 consecutive writes, which takes into account the different RLL encoding formats. (The complete derivation of this sequence, as well as description of the workings of the RLL encoding formats can be found in **Secure Deletion of Data from Magnetic and Solid-State Memory – by Peter Gutmann**)

Overwrite Data				
Pass No.	**Data Written**	**Encoding Scheme Targeted**		
1	Random			
2	Random			
3	Random			
4	Random			
5	01010101 01010101 01010101 0x55	(1,7) RLL		MFM
6	10101010 10101010 10101010 0xAA	(1,7) RLL		MFM
7	10010010 01001001 00100100 0x92 0x49 0x24		(2,7) RLL	MFM
8	01001001 00100100 10010010 0x49 0x24 0x92		(2,7) RLL	MFM
9	00100100 10010010 01 001001 0x24 0x92 0x49		(2,7) RLL	MFM
10	00000000 00000000 00000000 0x00	(1,7) RLL	(2,7) RLL	
11	00010001 00010001 00010001 0x11	(1,7) RLL		
12	00100010 00100010 00100010 0x22	(1,7) RLL		
13	00110011 00110011 00110011 0x33	(1,7) RLL	(2,7) RLL	
14	01000100 01000100 01000100 0x44	(1,7) RLL		
15	01010101 01010101 01010101 0x55	(1,7) RLL		MFM
16	01100110 01100110 01100110 0x66	(1,7) RLL	(2,7) RLL	
17	01110111 01110111 01110111 0x77	(1,7) RLL		
18	10001000 10001000 10001000 0x88	(1,7) RLL		
19	10011001 10011001 10011001 0x99	(1,7) RLL	(2,7) RLL	
20	10101010 10101010 10101010 0xAA	(1,7) RLL	MFM	
21	10111011 10111011 10111011 0xBB	(1,7) RLL		
22	11001100 11001100 11001100 0xCC	(1,7) RLL	(2,7) RLL	
23	11011101 11011101 11011101 0xDD	(1,7) RLL		
24	11101110 11101110 11101110 0xEE	(1,7) RLL		
25	11111111 11111111 11111111 0xFF	(1,7) RLL	(2,7) RLL	
26	10010010 01001001 00100100 0x92 0x49 0x24		(2,7) RLL	MFM
27	01001001 00100100 10010010 0x49 0x24 0x92		(2,7) RLL	MFM
28	00100100 10010010 01001001 0x24 0x92 0x49		(2,7) RLL	MFM
29	01101101 10110110 11011011 0x6D 0xB6 0xDB		(2,7) RLL	
30	10110110 11011011 01101101 0xB6 0xDB 0x6D		(2,7) RLL	
31	11011011 01101101 10110110 0xDB 0x6D 0xB6		(2,7) RLL	
32	Random			
33	Random			
34	Random			
35	Random			

Can overwritten data be recovered?

With the use of specialized equipment it is possible to detect the magnetic flux values stored on disk media with greater accuracy than is possible with the disk head assemble used in the drive itself. These techniques allow skilled technicians to retrieve data that has been deleted or overwritten.

MFM is a technique for imaging magnetization patterns with high resolution and minimal sample preparation. MFM uses a sharp magnetic tip attached to a flexible cantilever placed close to the surface to be analysed, and interacts with the stray magnetic field.

By using an optical interferometer or tunnelling sensor, the position of the cantilever can be measured, this gives a measure of the strength of the magnetic field acting on the cantilever as it is moved across the surface of the media.

Techniques such as MFM make truly deleting or overwriting data on media very difficult, the problem lies in the way that data is written to the media. When the write head writes new data to media it is not written to the precise location of the previous data, due to the inaccuracies in the head positioning system. This makes it possible to read traces of the old data along side the current track. (with the embedded positioning systems and extreme high densities of new drive technologies, this becomes less of a risk.)

Theoretically when a one is written to disk a one is recorded on the media, and when a zero is written a zero is recorded on the media. This is not the case is reality, the value is closer to 0.95 when a zero is overwritten with a one and 1.05 when a one is overwritten with a one. Normal disk circuitry is designed to accept both these values as one, but by using specialised circuitry it is possible to calculate what was previously written to the media.

The recovery of a least one or two layers of overwritten data can easily be done by reading the signal from the analogue head circuitry with a high-quality digital sampling oscilloscope. The output is then analysed by software to recover the previously recorded data. The software generates an "ideal" read signal and subtracts it from the actual signal, leaving the remnant of the previous layer. (with never channel coding techniques like PRML, the use of an oscilloscope to recover data is no longer possible)

Degaussing of the Media

A degausser is an external device that emits an alternating magnetic field that gradually decreases in strength. The result of this is to reduce the magnetic flux stored on the storage media to almost zero.

This is usually done by passing an alternating mains current through coils, thus generating an alternating magnetic field. The media is then moved through the magnetic field, first saturating the media and then gradually

reducing the magnitude to zero as the media is moved away from the magnetic field, leaving the media demagnetised.

For any external magnetic field to be effective in recording a signal on the media it needs be applied for a short period with a strength of 1/3 higher than the coercivity of the media. To effectively erase data on the media to the extent that recovery becomes uneconomical requires a magnetic field about five times greater than the coercivity of the media.

The table below lists the various coercivity levels of different media types:

Typical Media Coercivity Figures	
Medium	**Coercivity**
5.25" 360K floppy disk	300 Oe
5.25" 1.2M floppy disk	675 Oe
3.5" 720K floppy disk	300 Oe
3.5" 1.44M floppy disk	700 Oe
3.5" 2.88M floppy disk	750 Oe
3.5" 21M floptical disk	750 Oe
Older (1980's) hard disks	900-1400 Oe
Newer (1990's) hard disks	1400-2200 Oe
1/2" magnetic tape	300 Oe
1/4" QIC tape	550 Oe
8 mm metallic particle tape	1500 Oe
DAT metallic particle tape	1500 Oe

According to US Government guidelines the following classifications are made:

Class I <350 Oe
Class II 350-750 Oe
Class III >750 Oe

Degaussers are available for Class I and II but for Class III there are no degaussers available that can generate the recommended 7500 Oe to fully erase them.

This also creates a problem for hard disks as they have coercivity levels in the same order as Class III tapes. This makes degaussing hard disks a useless endeavour, this and the fact that degaussing would destroy the sync bytes, ID fields, error correction information, and other indicators needed to identify sectors on the media, thus rendering the drive unusable.

Physical Destruction or Physically Damaging the Media

Now that you are sufficiently paranoid the only real way to make sure data is gone forever is to destroy the media.

Physically disassembling a disk drive and removing the platters from the spindle is a highly effective form of protection. Despite claims to the contrary,

technology does not exist to remove the platters (without extensive control measures) from one device and read them back with another machine. At the time of manufacture, control signals (servo information) are written to every drive after it has been assembled. Any attempt to recreate or read back these signals once the exact alignment and relative positioning of the platters and the head stack have been altered is virtually impossible.

If the platters are removed - without strict engineering methodologies - the surfaces are useless for data recovery purposes.

Of course, once a platter has been physically removed, there is no reason not to have them simply scored with a single line to scrape the magnetic coating right off the platter. This would eliminate the one in a million chance that alignment in a new assembly is the exact same as the original.

Conclusion

It is effectively impossible to completely sanitise storage media by overwriting the previous data, no matter how many overwrite passes are made or what data patterns are used. However the use of these techniques, can make the job of an attacker far more difficult, if not prohibitively expensive. The only true way to made sure sensitive data is not recovered is to destroy the media or make sure the data is never written to disk in the first place. It is also a good idea to use encryption to protect data, so that if it is recovered it is still unreadable (make sure that the original unencrypted form can not be recovered otherwise this process is useless).

References:

1. Toxen, Bob. "Real World Linux Security : Intrusion Prevention, Detection, and Recovery" 1995
2. Gutmann, Peter. " Secure deletion of data from magnetic and solid - state memory." 22 July 1996. URL: http://www.fish.com/security/secure_del.html (20 February 2001).
3. Bundesamt für Sicherheit in der Informationstechnik. "Secure deletion of data media" 6 April 2000 http://www.bsi.bund.de/gshb/english/s/s2167.htm (20 February 2001)
4. Majors, Nicholas. "Data removal and erasure from hard disk drives." 1998 http://www.actionfront. com/dataremoval.html
5. Tao. "Secure file wiping" http://security.tao.ca/wipe.shtml
6. Tao. "Why normal delete is not sufficient" http://security.tao.ca/why - real-delete.shtml

Last Updated: July 21st, 2017

Upcoming Training

SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
Community SANS Omaha SEC401*	Omaha, NE	Aug 14, 2017 - Aug 19, 2017	Community SANS
Community SANS Trenton SEC401	Trenton, NJ	Aug 21, 2017 - Aug 26, 2017	Community SANS
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
Community SANS New York SEC401	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS