# Global Information Assurance Certification Paper

# Deleting Sensitive Information

Why hitting delete isn't enough

By
Hans Zetterstrom

Version 1.3

**Abstract**

This article intends to show that the deletion of files cannot be left to the delete key if those files are supposed to be disposed of securely. It proves how simply files can be recovered both under Windows and Linux if necessary security policy has not been extended to include the deletion of sensitive data. Popular techniques from each OS will be highlighted and procedures shown as to how to recover a deleted file. The article will then show how to securely delete a file so that current software tools cannot recover them. The article also touches on more advanced techniques beyond the means of most end users that can recover even the most securely deleted files, proving just how difficult it can be to remove data without leaving a trace of it behind.

**Introduction**

From failed .com pc liquidations to home users selling or giving away their machines most know that it isn't smart to leave personal information on the hard drive for the next owner to find and use as they see fit. Client lists, payroll information and company secrets all constitute things that even a failed company owes its former employees and clients to keep confidential. From the home side it can range from address books, to financial information. Be it corporate or home security, private data is something that should remain just that, and few have any wish for "old faithful" pc's to give up data they thought securely deleted.

Joan Feldman, president and founder of Computer Forensics in Seattle likens the PC to a continually running tape recorder [1]. It records every thought and every word we type into it, and often keeps numerous copies of those thoughts around for our protection. A quick look through some of the Windows folders will show a myriad of temporary files, each file storing session information, a snapshot of what is happening on the PC at a particular moment in time. Applications such as Word will auto save temporary versions of a document at regular intervals to save users the heartache of losing that important paper due to a sudden loss of power to the PC. What most users don't think about however is that very auto save feature can pepper the local drive with temporary versions of their paper that can be later used by someone else to recreate their thoughts. Not really significant if it is just a letter to family, but potentially embarrassing if that paper contains the salaries of the top twenty executives at your company, and now the person who found that temporary file and salvaged the data is sharing it readily with anyone who wants to see. Unfortunately just tracking down all the necessary files and assigning them to the trashcan is only akin to placing a veil over the data, it still very much exists.

## Out of sight, out of mind.

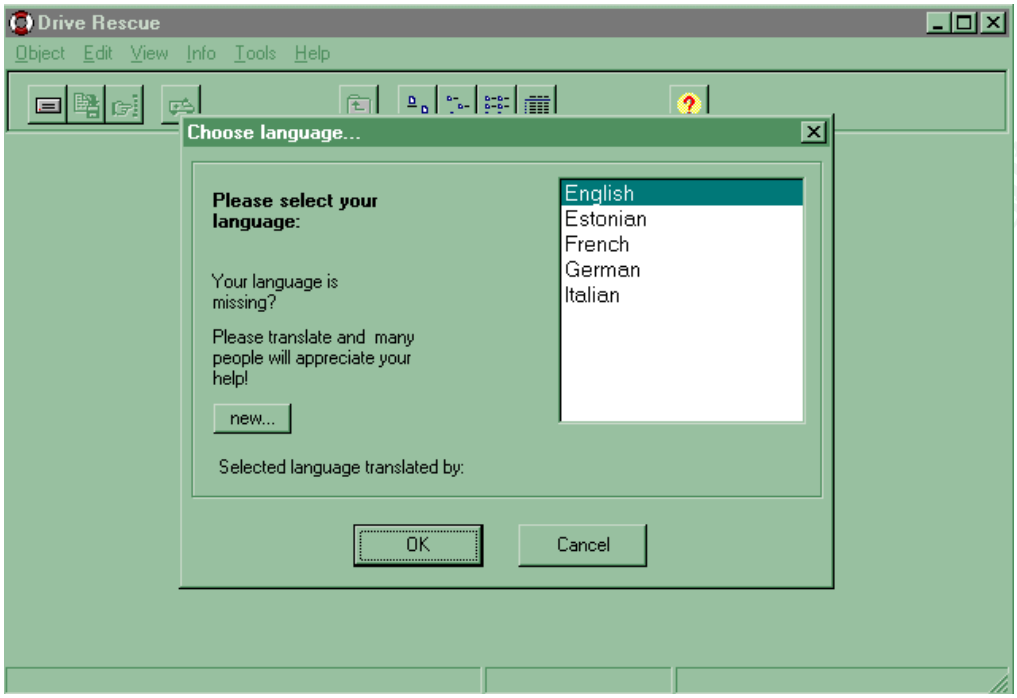
Under the Windows environment all the delete button really does is commit the file to the recycle bin, it still exists in its entirety at this point. Windows won't ever write over the data on the disk while it continues to exist in the recycle bin. Once the bin is emptied that data is marked as a candidate for overwrite. The data is never really removed from the hard drive at anytime, it just goes back into the pool of available disk space to be written to as and when necessary. To understand this better we have to look at how the Windows file system works on a basic level.

Upon saving a file, the file name, size and location of the file is stored in the File Allocation Table (FAT). When the file is deleted no data is touched, only the entry in the FAT is removed, which is the flag to tell the OS that it can reuse that disk space. However even once the OS is told it can safely write data to this space on the hard drive it does not necessarily mean that all the data will ever be overwritten. The reason for this is that comes down to the disk cluster size versus the amount of data being written to it.

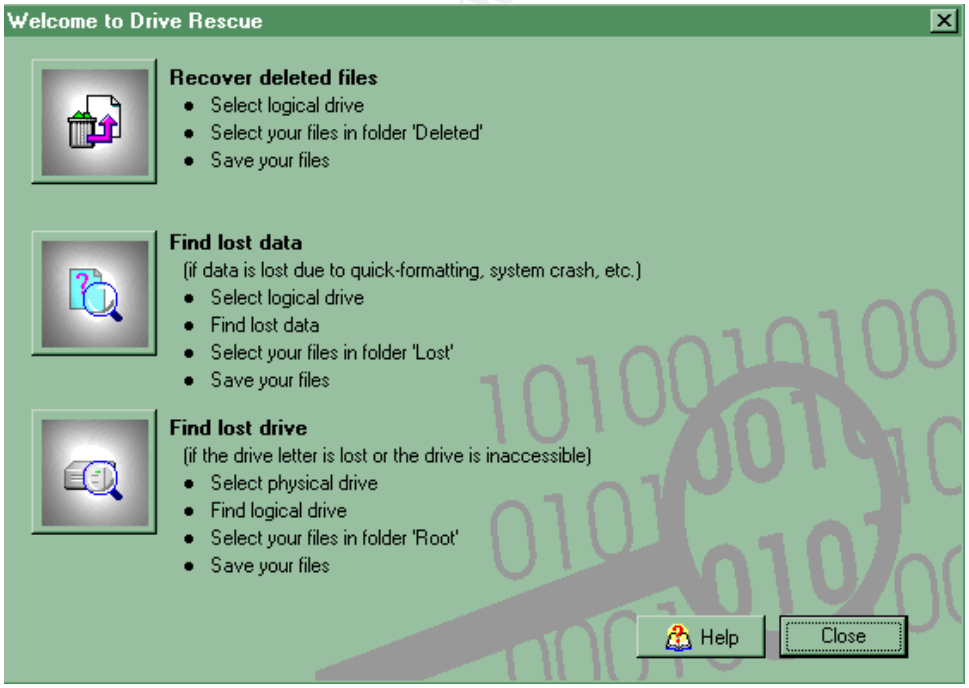
A cluster is the logical unit of file storage on a hard disk, which is managed by the computer's OS. Any file stored on a hard disk takes up one or more clusters of storage. The clusters associated with a file are kept track of in the FAT, so when you read a file, the entire file is obtained for you and you aren't aware of the clusters a file stored in, or their physical location on the disk. Under FAT32 the default cluster size is 4K regardless of the size of the disk but on older OS' a DOS limitation meant that as drives became larger and we chose to have larger partition sizes, the operating system had to increase the smallest possible unit of storage required by the data you were writing to your drive. Thus a large hard drive configured into a single or multiple large partitions would give away more capacity as slack space. Slack space is the difference between the size of the cluster and the size of the data being written into that cluster. Say for example that the cluster size is 8K and the data in the cluster is only 3K, the rest of the cluster will go unused, thus leaving 5K of slack space. Now imagine that previously that cluster contained 6K of information, only 3K of that information will have been overwritten by the new file, leaving the last 3K of the old data available for recovery.

## Drive Rescue for Windows

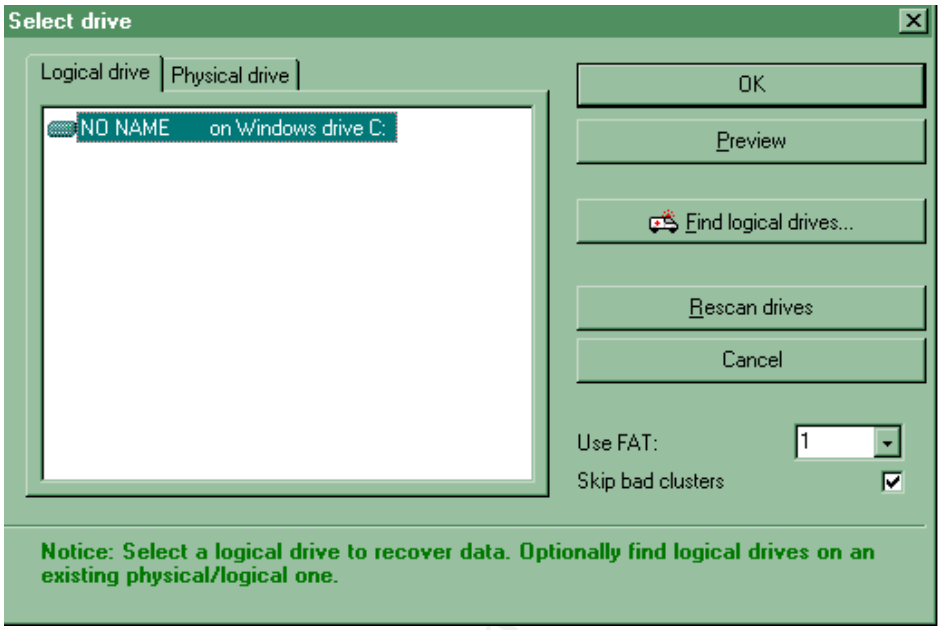
To illustrate how simply a file can be recovered using tools that are readily available, I downloaded a freeware utility called Drive Rescue, which is available from http://home.arcor.de/christian_grau/rescue/rescue.zip. On downloading and installing the application, it places a shortcut onto the desktop. Clicking on the icon your first decision will be to choose a language.



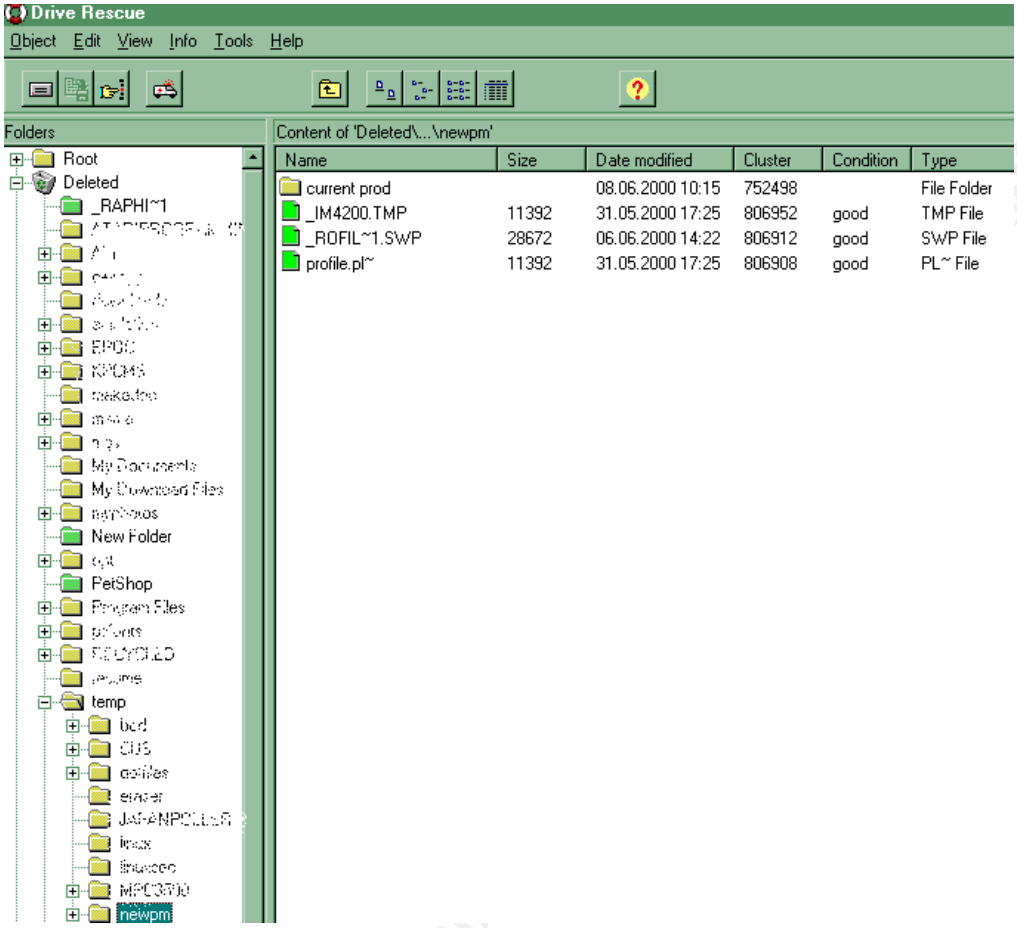

English works for me. Next, choose a task. The tasks range from finding deleted files, to those that have been lost due to formatting, or corruption or perhaps even a whole drive that due to errors cannot be mounted by the OS.



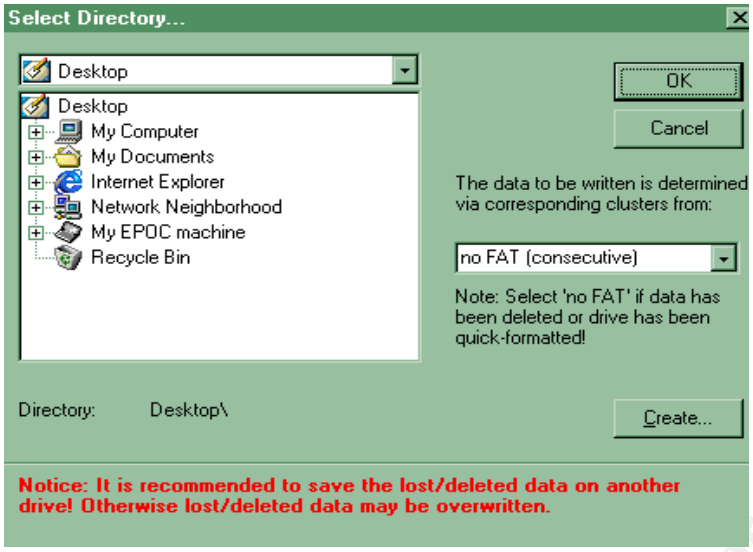
For this paper, I am concentrating on the first menu item, "Recover deleted files", so this is what I selected. Next step is to select the drive you want to work with. Many systems, as mine does, only have a single drive, so this selection is simple.




After selecting the medium to be scanned, **Drive Rescue** takes a few seconds to scan the drive for deleted data that is recoverable. The size of the drive determines how long this initial scan will take. Once the file list is created, the deleted files and folders will be highlighted in green. The navigation works in a manner similar to Windows Explorer. You must navigate to the directory structure that you last stored the file in, or you can use the Object->Find menu to do a text search for the name of the file that you are looking for. Since I know where the file was when I deleted it, I can navigate to it quickly using the Explorer menu. On the way I pass folders highlighted in green that indicate that I deleted them previously. **Drive Rescue** also gives information as to the condition of the files, the better the condition of the file; the more of the data will be recoverable.




The screenshot also shows the prevalence of both swap (.swp) and temporary (.tmp) files within windows. Both of these files could be restored and in the case of the file _IM4200.tmp it is possible to gain the exact file that profile.pl is, since this is an auto save version of the file. The swap file would be more complicated, but can also be used to extract the happenings on the PC at a given moment in time. Once the required file is found, it is a matter of right clicking the file and then selecting where to restore the file.




**Drive Rescue** warns that files should not be restored to the same drive as they are recovered from. This is because by writing to the drive, you may well be overwriting data that you later want to restore. Best practice would be to mount this drive to another OS and then restore files from this drive to another drive so that data is essentially read only.

**Drive Rescue** is not a particularly sophisticated software package, but it gives a clear example of how a PC user with a small amount of know how could gain access to sensitive information that the previous user thought was deleted.

## Linux Recover

So how do *nix based systems fair? Well the principles remain the same, the owner, group ownership, last read/write are preserved along with the data blocks. However these data blocks are going to be amongst the first reused by the OS. Although Unix based systems tend to err more on the side of good performance than easy recovery, this does not mean that it is particularly difficult to restore a file.

To illustrate the point I deleted two files from a Linux system and used the system as normal for a number of weeks. Using a tool called **Recover**, available from http://recover.sourceforge.net/linux/recover/download/recover-1.3b.tar.gz it was possible to restore the files to my /tmp directory. It is important to note that prior to Linux Kernel 2.2.x the indirect inode pointers were zeroed out when a file is deleted, thus limiting the ability to recover files to those where the connection between the file inode block and the data is preserved. The recoverable limit in Kernels prior to 2.2.x is 12*block size.

Since most Linux systems run many different services and it is constantly writing to the disk, it is important to unmount the partition quickly once it is decided to restore a file.

The fastest way to do this is to mount the partition as read only using the mount command,

<localhost>: mount –o ro, remount –n /home

Debugfs is a Linux file system debug utility that can be used to find the deleted inodes of files that have been removed from the system. It is passed the device name that the files were in when deleted and the argument required to find the deleted inodes.

<localhost>: echo lsdel | debugfs /dev/hda2

This will list all the available inodes, like this:

| Inode | Owner | Mode | Size | Blocks | Time |
|---|---|---|---|---|---|
| 12 | 0 | 644 | 628 | 1/1 | MAR SUN 3 12:39:17 2002 |
| 13 | 0 | 644 | 32 | 1/1 | MAR SUN 3 12:51:55 2002 |

So far, we know the inodes that we are interested in, but still have not recovered any data. A simpler way to discover the inode and restore the file is to use the recover script, which asks questions interactively to find the files that are to be restored. Recover requires that you know approximately when the file was deleted and it will return a list of available files.

<localhost>: ./recover

Recover v1.3b by Tom Pycke <Tom.Pycke@advalvas.be>

Scanning devices...
Ext2 devices:
recover: No valid standard devices found; are you a privileged user?

If your device is not listed, you can still use it
Please enter the partition's device name /dev/hda5
Getting inodes (this can take some time)...
debugfs 1.23, 15-Aug-2001 for EXT2 FS 0.5b, 95/08/09

In what year did you delete the file? (eg. 1999): 2002
In what month did you delete the file? (Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec or -1=unknown): Mar
On which day of the week did you delete the file? (Mon, Tue, Wed, Thu, Fri, Sat, Sun or -1=unknown): -1
What was the first possible day of the month on which you deleted the file? (1 - 31): 2

What was the last possible day of the month on which you deleted the file? (1 - 31): 4
What was the soonest possible hour(0-23) when you deleted the file? 7
What was the latest possible hour(0-23) when you deleted the file? 17
What was the soonest possible minute(0-59) when you deleted the file? 0
What was the latest possible minute(0-59) when you deleted the file? 59
What was the minimum possibly file size in bytes? (0-2147483640): 0
What was the maximum possibly file size in bytes? (0-2147483640): 20000

What was the user ID of the deleted the file? (-1 if you have no idea): 0 (root)
=> 13 32 MAR SUN 3 12:39:17 2002
=> 12 628 MAR SUN 3 12:51:55 2002

2 inodes found. Where shall i dump them? (directory): /tmp

Please type some text the deleted file should have included (type: * if you don't know it):
*
Please wait...
Dumping inode 13 to /tmp
Dumping inode 12 to /tmp

Linux is definitely not as easy to restore files on as Windows, it is however possible to restore information that has been deleted and has not yet been overwritten.

**Scanning Tunneling Microscopy**

The easy solution to file restoration is to defragment and format the hard drive thus deleting all the data that is on it, correct? Well no not necessarily. There is a variety of documentation available proving the ability to find information buried beneath a Solaris install that was from a prior install of Windows. However once the trail on the data has become this faint, it is unlikely it can be salvaged in its entirety and the time taken to restore is very great. Using a method called Scanning Tunneling Microscopy, approximately 50 bits of data are read at a time and each 50-bit scan takes about five minutes. If you are talking about a 10 Gigabyte drive, storing about 80 billion bits, that is a great deal of time it would take to scan that one drive alone. Peter Bedrossian states that, "The scanning tunneling microscope provides a picture of the atomic arrangement of a surface (of a disk) by sensing corrugations in the electron density of the surface that arise from the positions of surface atoms. A finely sharpened tungsten wire (or "tip") is first positioned within 2 nanometers of the specimen by a piezoelectric transducer, a ceramic positioning device that expands or contracts in response to a change in applied voltage." Based on this highly specialized technology and the intensely skilled labor required to use this technique, it is safe to assume that the average Joes of this world don't have much to be concerned about. On the other hand however, it is just as safe to assume that

companies such as Arthur Andersen, Enron and before them Microsoft are now very well aware that the data that they previously held precious but deemed 'surplus to requirements' can be recreated.

Just as in the home user examples of file restoration, the experts never work on a live file system, but take copies of the disk to work on those instead. Joan Feldman best sums up the reason for this, "If, in addition to examining files directly on the computer, I open it up to read it, I have changed the meta-data for that file. That changes the last access date and time and if I do anything more I may also have modified that file. So, it becomes very difficult to weed out or parse through that which was there prior to the review. If you have to testify to it, you wind up dancing through a sea of razor blades and you start to look like an idiot." [1]

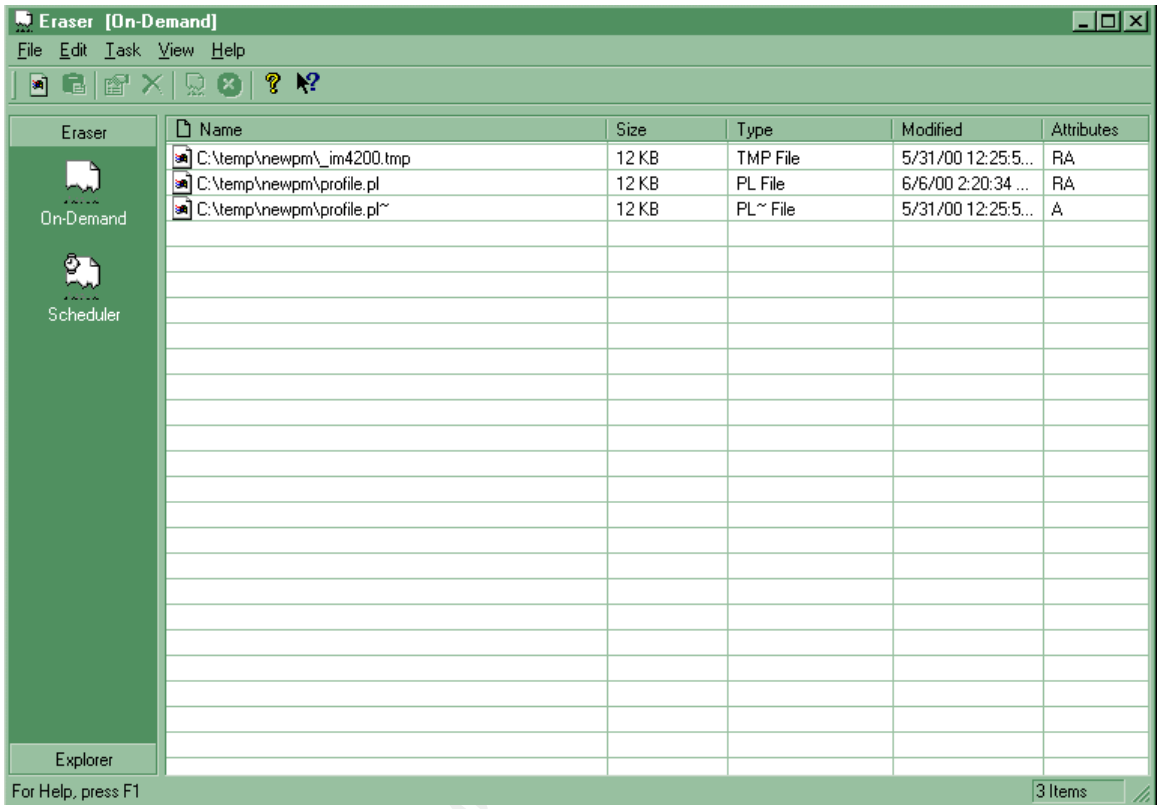
## So how DO you remove data permanently?

The short answer to this is that you don't. Government requirements for highly secure data is that when a hard drive is deemed end of life, it either has the platters sanded down or is dissolved in acid to leave it useless for data gathering purposes. Overwriting data once is not usually good enough to prevent data recovery, instead it is recommended that a minimum of three passes are made writing alternating zero and one patterns over the data and then further passes with random data, the more passes the better the chance that no data can ever be recovered. The approach recommended by Peter Gutmann, author of "Secure Deletion of Data from Magnetic and Solid-State Memory" is a total of thirty-five passes. It is said that this method is the minimum that will guarantee that every spot on the hard drive is written to and can even prevent what is known as "Ghosts" on the hard drive. Ghosts are residual traces of data, caused by the fact that like charges used to write the binary data to the hard disk repel one another. This means that (assuming 0 is a "-" charge and 1 is a "+" charge) when alternating positive and negative charges are written next to one another the width of those bands is greater than if it had been written next to another positive band, because opposites attract [12]. From this computer forensic specialists can attempt to recover some data on the drive despite it having been overwritten multiple times.

## Eraser for Windows

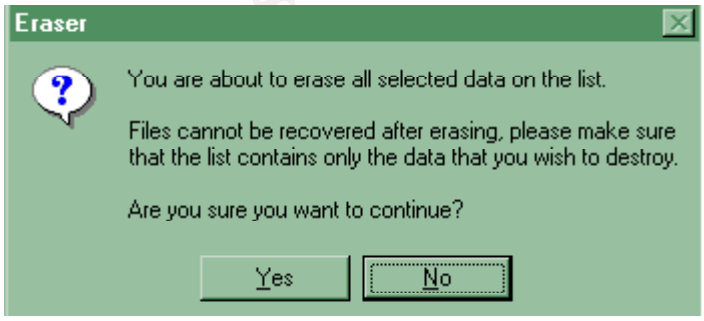
There are programs available that will help perform the multiple overwrites that are desirable for secure data removal. On Windows **Eraser** is just such a tool, which is available from http://prdownloads.sourceforge.net/eraser/eraser53.zip. After download and install it was a simple matter of adding the files to a list of tasks and then running that task at the level of deletion required. The levels of deletion range from one pass to the

three and seven passes recommended by the Department of Defense, through to the thirty-five passes recommended by Guttman.

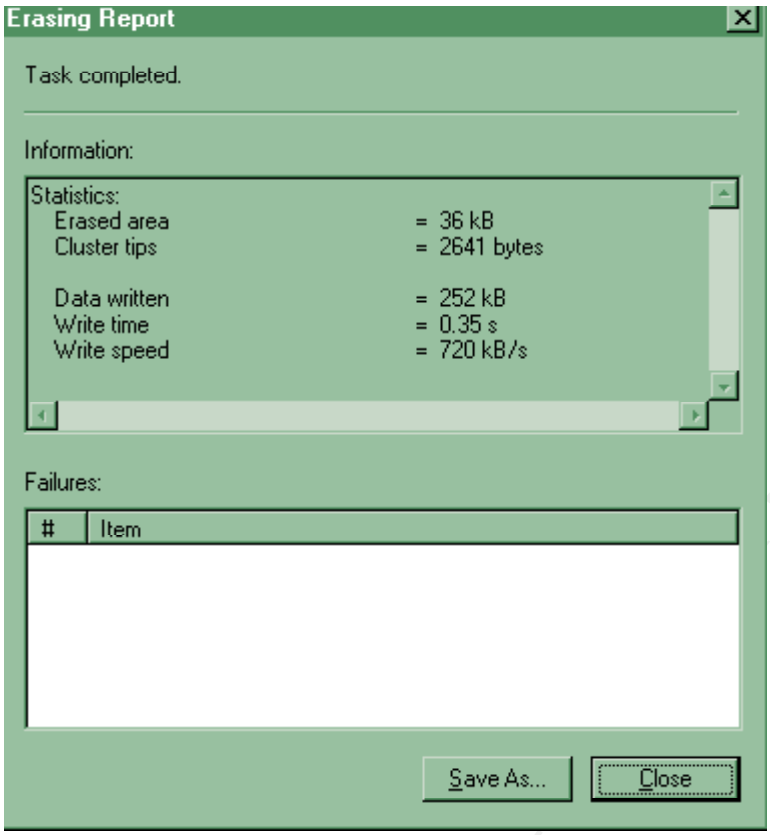



To add the files, simply select the File->New Task menu item and browse to the file required for deletion. In the above example I selected the three versions of the file that **Drive Rescue** was able to restore in my earlier test.

Selecting the Task->Run All menu item brings up a warning screen.




Affirming this screen causes the task to run and display the results of the run.




For my test, I chose the US Department of Defense recommended seven pass deletion level. A follow up test using the **Drive Rescue** tool shows a slew of deleted zero length scratch files of poor quality in the directory where the files I deleted once lived, but no record of the actual files themselves can be found. The scratch files contain no usable information at all indeed even their creation date is unobtainable by **Drive Rescue**.

Content of 'Deleted\...\newpm'

| Name | Size | Date modified | Cluster | Condition | Type |
|---|---|---|---|---|---|
| current prod | | 08.06.2000 10:15 | 752498 | | File Folder |
| _ | 0 | 00.00.1980 00:00 | 0 | poor | File |
| _ | 0 | 00.00.1980 00:00 | 0 | poor | File |
| _ | 0 | 00.00.1980 00:00 | 0 | poor | File |
| _ | 0 | 00.00.1980 00:00 | 0 | poor | File |
| _ | 0 | 00.00.1980 00:00 | 0 | poor | File |
| _ | 0 | 00.00.1980 00:00 | 0 | poor | File |
| _ | 0 | 00.00.1980 00:00 | 0 | poor | File |

If I only want to ensure that all the space on my drive that is marked as free was sanitized then **Eraser** can assist me with that also. I add a new task to erase all free space on drive C:\ and run that task. The question is since I have not defragmented my hard drive in a while would it make sense for me to defragment first before I wipe? The answer to that would be no. When defragmenting, the OS will attempt to reassign data consecutively on the disk thus risking that data is assigned to a cluster of a lesser size than occupied it previously, leaving data in the slack space of that cluster that can be retrieved. To visualize say there are four files:

| Filename | Size | Starting location | Comment |
|---|---|---|---|
| Foo.txt | 200 | 500 | |
| Bar.txt | 200 | 700 | Sensitive file |
| Fez.txt | 100 | 900 | |
| <free> | | 1000 | |

Now you deleted the file Bar.txt a while ago, without wiping it as you went. After defragmenting the disk layout would like this:

| Filename | Size | Starting location | Comment |
|---|---|---|---|
| Foo.txt | 200 | 500 | |
| Fez.txt | 100 | 700 | |
| <free> | | 800 | |
| | | | |

The problem with this is that since the space where Bar.txt existed was not securely wiped prior to Fez.txt moving to that disk location it is possible for data to be recovered, albeit difficult and resulting in partial data being recovered. So, recommended procedure is to wipe the free space first before defragmenting the drive.

To wipe the swap file space within Windows is a little trickier, since tampering with the swap file while Windows is running can cause it to crash. Although some tools claim to be able to wipe an active systems swap file, Sami Tolvanen the author of Eraser disagrees,

"There are applications that claim to overwrite swap file contents while Windows is running. They are usually trying to accomplish this by allocating huge amounts of memory and hoping that the operating system will write it to the disk (overwriting previous data). Doing this may even prove to decrease security instead of increasing it - instead of flushing the memory allocated by the overwriting program to the swap file, Windows may as well decide to save the memory allocated by some other application to the disk, possibly causing sensitive data that otherwise would have remained in the memory to end up on your drive. And even if the user is real lucky and everything goes as planned, the data currently allocated in the swap file still cannot (and will not) be

accessed." [4]

Fortunately Eraser provides a DOS utility that will wipe the swap space for you, so boot the PC with a DOS boot disk and run the tool **eraserD**, which is **Eraser** for DOS. For people who don't mind badly degraded performance, Windows can run without a swap file, however this is not recommended for most PC users who want more than one application open at once, since Windows uses the swap file pretty heavily on systems with limited memory. Eraser also allows the scheduling of data to be wiped automatically, so wiping that free space which can take a long time when set on the thirty five passes setting can be set to do so at 1am while you sleep. Additionally Eraser supports overwriting with Pseudo Random data, which is based on ISAAC a fast cryptographic random number generator that generates 32bit random numbers. There is one gotcha to all of this praise for **Eraser** and other Windows based products like it, on NTFS systems there is a flaw. Quoting from the Kurt Seifried security advisory [10],

"In the NTFS file system a facility exists to bind additional data to a file or directory, called an alternate data stream. These alternate data streams cannot be be [sic] removed, unless the parent file or directory is destroyed. Unfortunately most file wiping utilities only deal with the primary data stream and do not wipe the alternate data streams, thus leaving data intact."

According to the advisory every software package that was tested, failed with regard to the deletion of alternate data streams. Thus say thumbnail images of larger images that have been deleted remain behind they must be sought out by hand and removed securely.

## Regcleaner for Windows

Windows takes an additional tool to clean up the registry. As most users know, almost every application litters the registry with settings and information and sometimes this information needs to be removed easily and securely. **Regcleaner**, available from http://www.vtoy.fi/jv16/programs/RegCleaner.exe does this easily by reading your registry and then offering to remove entries based on their application affiliation, file type, uninstall menu and startup menu. It is common for applications to store their passwords and other sensitive data hashed in the registry, and sometimes this data is then left behind when the application is uninstalled. Regcleaner can help to track down this data and remove it before unauthorized access is made.

## Linux Secure Delete

Under Linux the tool **Secure Delete** performs a similar task to Eraser, available from http://freshmeat.net/redir/securedelete/9377/url_tgz/download.php?t=r&d=secure_delete-2.3.tar.gz. Under standard configuration, the srm tool (secure rm) will delete with 38

passes. These passes break down like this (from the Secure Delete README),

1x overwrite with 0xff
5x random passes
28x overwriting with special values to make the recovery from MFM and RLL encoded hard disks almost impossible
5x random passes

Needless to say with this level of file removal, it was impossible for me to restore the files I deleted with this tool. Wiping of free space is done with the sfill (secure fill) tool that will fill the free space on the disk with 38 passes of information. According to the author of the tool, it is the special values used during the middle 28 passes of each wipe and fill routine that ensures that the data cannot even be recovered by Scanning Tunneling Microscopy methods. The smem and sswap will securely erase entries remaining within the RAM and swap areas of the machine. One requirement of truly securely deleting files from within Secure Delete is to have /dev/urandom available since the included randomization is not as robust as /dev/urandom. Finally the real beauty of this tool is the simple rm.diff file that is included with the source that allows you to recompile the rm source to automatically delete everything securely when you type in rm. You could also just alias rm to srm, either way they both work equally well.

## In Conclusion

Secure deletion of files and swap space is good practice if you value your private data. However it is only as good as the security of your PC both physically and from access to it from the network you are on. Files may well exist on backups that are out of your control and it is important to know how those backups are made, how long they are kept and that you trust where they are being stored. It is worthless to have an important secret document on your hard drive and assume it is safely destroyed only to find out that the nightly backup tapes were stolen from the back of the system administrators car. As important as a good backup policy is to a company, so it should have a policy that deals with the deletion of data that is deemed sensitive. Frequently this information is left to the discretion of the individual within the company and this can lead to embarrassment especially if important data should not have been deleted and was. Christopher Wolf, an attorney at Proskauer Rose dealing with cases involving the destruction of documents states, “clients should keep items that they know may be needed in an investigation” [1]. Indeed it is illegal to delete data when under a subpoena or threat of one, and a Judge can instruct a jury to assume that the documents that were destroyed included information that would be damaging to the individual or company involved. This could be the difference between a company reputation being forever tarnished and the ability to use that document to prove their innocence.

For the average user a good file deletion utility can provide some peace of mind and ensure that when the time comes, no unwanted eyes can view deleted data, since it is unlikely that anyone would want to spend the sums of cash required to finance a Scanning Tunneling Microscopy discovery project on any of our drives. For those that have stepped into areas where someone may want the data bad enough to spend the time and money, they best have the vat of acid standing by, or at least a strong belt sander since complete guarantees of data removal are very hard to find.

Many Thanks to Kenda Stellard for taking the time to proofread this article.

## Resources


[1] Paul Festa and Lisa M. Bowman, Computer hinder paper shredders, 4 February 2002, URL: http://news.com.com/2100-1023-829004.html

[2] Associated Press, Sleuths probe Enron E-mails, 16 January 2002, URL : http://www.wired.com/news/print/0,1294,49774,00.html

[3] Rovert Vamosi, I know what you did on your PC last summer, 16 October 2001, URL: http://zdnet.com.com/2100-1107-504091.html

[4] Sami Tolvanen, Eraser, URL: http://www.tolvanen.com/eraser/faq.shtml

[5] Alexander Grau, Disk Rescue
URL: http://home.arcor.de/christian_grau/rescue/index.html

[6] Jouni Vuorio, Regcleaner, URL: http://www.jv16.org

[7] Tom Pycke, Recover, URL: http://recover.sourceforge.net/linux/recover

[8] Van Hauser , Secure Delete,
URL: http://freshmeat.net/projects/securedelete/?topic_id=43

[9] Peter Gutmann, Secure deletion of data from magnetic and solid-state memory, 22 July 1996
URL: http://www.cs.auckland.ac.nz/~pgut001/secure_del.html

[10] Kurt Seifried, Multiple Windows file wiping utilities do not properly wipe data with NTFS file system, 21 January 2002
URL: http://www.seifried.org/security/advisories/kssa-003.html

[11] Anton Chuvakin, Ph.D , Linux data hiding and recovery, 10 March 2002,
URL: http://www.linuxsecurity.com/feature_stories/data-hiding-forensics.html

[12] Ghosts, http://security.tao.ca/ghosts.shtml

[13] Peter Bedrosian, Scanning Tunnel Microscopy,
URL: http://www.llnl.gov/str/Scan.html