Secure Delete

Merely deleting files is not good enough if you wish to cover your tracks and maintain your privacy. Files must be physically overwritten, a process known as wiping, in order to prevent their discovery and resurrection from recovery programs such as Recuva and Photorec.

The **secure deletion toolkit** is an essential set of command line tools that allow you to wipe files, memory, the swap space, and free space to physically destroy existing contents beyond recovery by forensics software.

Installing Secure-Delete

All secure delete tools are freely available in the Ubuntu repository under the package **secure-delete**. To install, open the Synaptic Package Manager, enter secure-delete in the Quick search box, locate secure-delete, right-click it, and choose **Mark for installation**. Click the Apply button at the top.

To install from the command line, enter,

```
sudo apt-get install secure-delete
```

What's Included?

There are four separate program that constitute the whole package, and each performs a different type of wipe operation.

```
srm      wipe existing files
sfill    wipe free space
sswap    wipe the swap space
sdmem    wipe the RAM
```

srm – Secure rm

This deletes files like the **rm** command, but it does so by overwriting the file and its inode with random bytes.

```
srm file
```

Each overwrite is called a pass. By default, the Gutmann 35-pass method is used, but this might be overkill. In my experience, one pass of random data wipes files to the point where neither Recuva or Photorec can recover them. The larger the file, the longer it takes to wipe it.
However, you may adjust the number of passes to shorten the wiping time. See **man srm** for details about which options to set. One quick but useful operation is this:

```
srm -vrll directory
```

This deletes files and directories recursively with one random pass and displays progress in the terminal, something useful for large files to avoid wondering if the computer froze or not. In my tests to recover deleted files using this method, the single, random pass sufficiently wiped all files to the point where not even Photorec could recover them.

Your security needs might vary, so take into account who and what you are trying to protect yourself against and choose the best wiping method for yourself.

sfill – Wipe Free Space

No matter what file system is used, files remain on hard drives, USB devices, and flash media long after deletion. Deleting a file only marks its data blocks as free for use; it does not actually remove the file from the media. The file's directory entry is gone, but its contents are still there.

Until the file's data blocks are overwritten with new or random data, they exist for recovery. All data blocks of a file must be overwritten to wipe the file and prevent its

recovery. Even partially recoverable files can provide clues about your past computer usage.

All deleted files exists in free space, which is the space on a media available for file storage. Free space does not mean blank space. Partial and impartial files continue to exist in the free space, but there is no way to wipe the files since they do not exist to the operating system. Thus, srm cannot be used to wipe them.

This is where sfill enters the picture.

```
sfill .
```

(Include the dot '**.**' to indicate the current directory.)
sfill works by creating a very large file (named oooooooo.ooo) of random data that fills all available free space. Doing this overwrites everything in the free space but leaves existing files alone. Once all free space is exhausted, sfill deletes oooooooo.ooo and the free space is recovered.
Keep in mind that you might need to run this as root **sudo sfill .** to gain access to all areas of the hard drive, but if not, at least you can delete the free space from the area your user account has access to.

sswap – Wipe the Swap Space

The swap area used for virtual memory is beyond the touch of users. Just like deleted files, data swapped into and out of the swap space persists in the swap area. To wipe the swap area, use sswap, but disable the swap area first.

1. Find the swap partition.

```
sudo fdisk -l
```

The swap partition will read **Linux swap** in the listing. Find which device contains the swap space. It will read something like /dev/sda5 or /dev/sdc2.

2. Disable the swap partition by its device

```
sudo swapoff /dev/sda5 (or whatever device was found in step 1)
```

3. Wipe the swap space by its device (one-pass random for speed)

```
sudo sswap -vll /dev/sda5
```

4. When finished, enable the swap partition

```
sudo swapon /dev/sda5
```

Wiping the swap space is more involving than wiping a file, so use it if you think you need to be extra cautious. Like srm and sfill, **man sswap** provides a list of options.

sdmem – Wiping RAM

Believe it or not, there are actually elaborate ways to recover data from physical RAM modules. Yes, if performed properly, RAM contents can be dumped into another machine for analysis. Let's thwart that technique by wiping RAM.

```
sdmem -vll
```

This will produce a terminal filled with asterisks **** to show activity while the free space in RAM is being overwritten. The computer is still functional while the RAM wiping is occurring.

Caveats

Wiping data is never a 100% recovery-proof technique. Even the secure-delete man pages stress this point and provide cautions. One reason involves the robustness of today's journaling filesystems.

Unlike FAT, journaling filesystems, such as ext3, ext4, and NTFS (to name a few), are designed to recover data and reduce errors in the event of hard drive failures, power failures, and computer crashes. Many incorporate some level of redundancy and often store metadata (data about the data) that might be unaccessible for wiping. Metadata and file tips (unused data at the end of a data block) might contain scraps of information that provide clues about what was stored in that area.

There is no guarantee that all data will be overwritten in all locations on the media and in its cache. However, wiping files is a good practice, and, at the very least, it prevents consumer-grade recovery techniques and programs from accessing your private files.

I have tried a variety of programs to recover data deleted with the techniques listed above, and not one succeeded, so at least the average user (which includes the majority of computer users who do not have access to specialized data recovery hardware) is unable to recover the wiped data.

Other thoughts:

- If you use a shared computer, wiping your personal files will help protect your privacy.
- Did you purchase a new SD card for your digital camera? What will you do with the old one? Give it away? Throw it away? Not so fast. No matter how many times you delete or format the SD card, your old pictures will remain. Wipe it first. The secure-delete tools will do an excellent job.
- Wipe your hard drive before disposing it. You never know where it might turn up. Some wiping is better than none, and wiping free space helps prevent others from reading your deleted emails (yes, it is possible).

Scripting

“I don’t like to type in a terminal. Can I write a wipe script that is executable from the Nautilus context menu?”

Yes, but be warned about choosing the wipe script by mistake. You could accidentally wipe files by accident. If you do use a script, add confirmation protections using Zenity to confirm before wiping.

Conclusion

Wiping files is only one link in the computer security chain, but an important one. While many other software packages are available, the secure delete toolkit is freely available, simple to install, comprehensive, and easy to use.

Source : https://delightlylinux.wordpress.com/2012/06/14/secure-delete/