Shred files and wipe disks

12-15 minutes

Normally when software deletes a file, only the “metadata” is erased: that means the complete contents often can easily be recovered, so BleachBit (and similar applications) offer secure erase features (also called secure wipe, secure overwriting, or file shredding) to permanently remove data. Some applications even advertise “advanced” erasure methods referencing important names in security such as Gutmann, the United States Department of Defense, and the NSA, but these references often mislead people to waste time on snake oil technological remedies while ignoring important basics. Any product or method suggesting a convenient, comprehensive solution to security is deceptive: convenience and security oppose each other. This guide will explain **how 1 pass is enough, but 35 passes are not enough.** Regardless of the tools you use, please read this guide carefully and completely.

Myths and legends

Most of the confusion regarding the topic of data remanence (data left behind after it is deleted) is because of myths and urban legends. Before discussing what is true, let’s preview what is not:

- False: Data on a hard drive overwritten by one pass can be recovered by powerful government agencies
- False: Overwriting data with multiple passes makes it harder to recover than overwriting it with a single pass.
- False: Peter Gutmann thinks data should be overwritten with 35 passes to prevent recovery.
- False: Peter Gutmann’s paper applies to modern hard drives.
- False: There are methods approved by the DOD (5220.22-M), NSA, and Gutmann to shred files.
- False: The United States Department of Defense approves of overwriting of a whole hard drive as a data sanitation method.

The details are explained below.

What features does BleachBit have to securely wipe files?

Today BleachBit offers these features

1. Overwrite specific files found by its cleaners to hide the contents of these files (such as Firefox Internet cache).
2. Overwrite specific files found anywhere on the hard drive (such as a confidential spreadsheet on your desktop).
3. Wipe free disk space to hide the contents of files previously deleted by any software.
4. Wipe memory and swap to wipe data stored in RAM such as passwords and web pages (currently, only on Linux).

Shredding is much slower than deleting because deleting changes only the file system metadata, which is small and a consistent size for all files: the metadata is the name of the file, its location on the disk, a time stamp, owner, etc. On the other hand, shredding takes time proportional to the size of the file.

Because additional passes add no value and only offer a false sense of security (see below), BleachBit does not implement multiple passes. When overwriting is enabled, BleachBit wipes the data with a single pass of blank data. At the end of a file, it may write additional data to wipe the slack space. To hide the original name, BleachBit renames the file to a long name and then a short name. Finally, the file is deleted.

Unlike other cleaner applications, BleachBit shreds Internet history in Firefox without deleting the whole Firefox Places database.

Are these features secure?

Concerned about the security of my house, once I asked a locksmith whether I should upgrade the standard lock on my front door. Not a good salesman, he replied, “Why bother? A burglar would just break the window.”

Is driving a new car with air bags and a good crash test rating safe? Probably. How about when the car is operated by a 15-year old on a busy road at night while texting on his cell phone? Probably not. This analogy demonstrates two things. First, security is not black or white: there is always risk ranging from near (but not completely) zero to near 100%. Indeed, sometimes people even walk away from catastrophic car accidents. Second, the context is important: what is good in one situation is not good in another. Therefore, a better question is…

How secure are BleachBit’s methods?

When used appropriately, BleachBit’s data wiping features are generally enough to hide traces of most data from most people.

Even more important than asking how secure an application or technique is, start by asking yourself

1. What am I trying to hide? Is worth a lot of money? A few thousand that is in your bank account? A one million dollar trade secret?
2. How much harm could it do if disclosed? Embarrass someone? Hurt a relationship? Break compliance with HIIPA or SOX? Bankrupt a large company? Shift global political power?
3. Who would try to find it? A family member? A powerful rival company? A superpower government?
4. How much time, money, and skill does that entity have? How much is it willing to spend to find your secrets?

The answers to these questions will lead you to the appropriate level of caution.

Are multiple passes better than one pass?

Some applications offer “advanced” and “high security” erasure techniques such as Gutmann method (35 passes), Department of Defense (DOD) standard (7 passes), National Security Agency (NSA) “approved” (3 passes), etc. Introduced by a poor reading of Peter Gutmann’s ancient paper, people incorrectly believe that overwriting the same data multiple times makes it more difficult to recover. Years after his original paper, Peter Gutmann himself tried to clear up the confusion caused by his original paper [*]:

> Some people have treated the 35-pass overwrite technique described in it more as a kind of voodoo incantation to banish evil spirit. […] In fact performing the full 35-pass overwrite is pointless for any drive since it targets a blend of scenarios involving all types of (normally-used) encoding technology.

Wikipedia explains further that a single pass is enough [*]:

> The chances of overwritten data being recovered from a modern hard drive amount to “urban legend”. He also points to the “18 1/2 minute gap” Rose Mary Woods created on a tape of Richard Nixon discussing the Watergate break-in. Erased information in the gap has not been recovered, and Feenberg claims doing so would be an easy task compared to recovery of a modern high density digital signal.

If recovery of data on modern media overwritten by a single pass were possible, scientists would publish papers about it, data recovery companies would charge top dollar for it, the use of it by government organizations would leak out in court cases, and drive manufactures would exploit it to increase storage capacity..

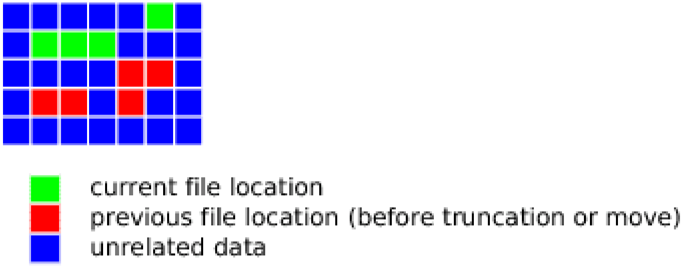
Second, shredding software which advertises these features often glosses over the application of the technique. **The DoD 5220.22-M standard was never intended be shred individual files or to wipe free disk space**: it was intended to wipe the entire hard drive causing a complete data loss including the operating system and all software, settings, and documents. Also, Department of Defense approves software shredding techniques only within the DoD: for storage devices released out of the Department of Defense, only mechanical destruction is approved. In other words, any software which shreds individual files or free disk space cannot be DOD or NSA compliant.

Limits of shredding files and wiping free disk space

Shredding individual files and free disk space has limited benefits for any cleaner application, including BleachBit. Once you understand the limits, you will know whether taking extra mitigation steps is worthwhile.

Shredding an individual file properly assumes its location can be completely known, but basically it can only be known in one ideal case. The ideal case has three characteristics:

1. The file size has never shrunk because of editing. Imagine starting with a 3MB spreadsheet, editing it down to 1MB (using the spreadsheet application), and asking the cleaner application to delete the 1MB version: the cleaner has no way of knowing where the missing 2MB was allocated on the physical hard drive. (Remember: file systems often don’t store files continuously, so you can’t assume the missing part was directly after the known part.)



2. The file never moved. Imagine the spreadsheet software saves the document by writing a new copy to a temporary file, deleting the old copy, and renaming the temporary file to the original name. In this case, the cleaner application has no way of knowing where any of the old spreadsheet was located.

3. The file system overwrites files to the same place. This is a good assumption. On Windows NTFS and on Linux the most common ext3 configuration (which is the default on Ubuntu 9.10 and other Linux distributions) overwrite files in the same place, but transparent disk compression, encryption, and sparse files may not overwrite files in place.

Though BleachBit cannot know the location of file after it is deleted or moved, neither can any other software. The deleted data is now floating in a giant pool of noise. In other words, a file shredded by BleachBit *even in these non-ideal scenarios* is difficult to recover *partially* and likely impossible to recover *fully*. A file shredding in the ideal case should be impossible to recover—even partially.

The three problems above are addressed by wiping free disk space: it doesn't matter where the previous file was located. If the deleted file is allocated now by a new file, the new file has overwritten it. If the previous location is not allocated now, the cleaner will overwrite it.

However, wiping free disk space has several of its own challenges:

1. It can be slow, so many people are not willing to use it.
2. File systems allocate space in fixed chunks called a block size, and many files do not use all the last block. A 5,000,000 byte file on a 4096 size block file system would use 1220 full blocks and 1 partial block with 2880 bytes. Say the file was deleted and a new file in the same place used 1024 bytes of the last block. That means 1856 bytes of the old file (0.03%) is not overwritten in what is called the "slack space" of the new file. Because cleaning slack space is tricky and realistically little useful data can be recovered from such tiny pieces (typically not more than 4096 bytes), BleachBit does not clean slack space when wiping free disk space. (Remember: BleachBit *does* wipe slack when wiping individual files.)
3. When an area of a modern hard drive is damaged, it automatically remaps the bad sector to a spare. The operating system and applications are unaware of the move, so wiping the drive ignores the damaged area. According to DBAN, a powerful disk wiping tool, it does not erase remapped sectors and hidden areas.

How to securely delete data

To permanently delete data, there is an order of progression with trade-offs of convenience and time vs privacy:

1. Shred the file (with one pass).
2. Overwrite the free disk space.
3. Overwrite the whole drive (including the operating system and all data).
4. Mechanically destroy or degauss the drive.
5. Destroy data on backups, ISPs, online accounts, etc.

However, in practice things become more complicated.

Keeping data private

Here are some suggestions to keep your data private

1. Don't keep secrets. It's easier to sleep.
2. Don't waste time with multiple passes for data sanitation.

3. Second guess any software which advertises multiple passes to wipe files or free disk space. Do the authors not honestly know what they are doing, or is a useless feature for marketing purposes?
4. Use full volume encryption, though someone may hit you with a $5 wrench until you reveal the key.
5. If giving a hard drive (or whole computer) to someone else, use DBAN to wipe the entire drive, including the remapped sectors—even though reinstalling an operating system, security updates, applications, and settings is a pain. It's not enough to delete files, empty the recycle bin (or trash can), and wipe the free space because some useful data may be in the swap file, hibernation file, Windows registry, and application registries (such as passwords in Firefox's configuration). If you are not willing to do that, minimally delete the user accounts on the system and *then* wipe free disk space.
6. If you need DoD class security, use the only sanitation method approved by the DoD 5220.22-M standard: degauss or mechanically destroy the storage device.
7. Don't assume you control all the data. Say you download a file from www.example.com: there may be records on your computer, your ISP, www.example.com's server, www.example.com's ISP, www.example.com's backup site, the Internet backbone, etc. Think about how much data is stored on your email server, Facebook account, etc.
8. Don't use any computers because the Nosy Secret Agents may looking over your shoulder using Van Eck phreaking.

Suggested reading

- "Data Remanence" (Wikipedia)
- "Gutmann method: criticism" (Wikipedia)
- One big file is not enough: A critical evaluation of the dominant free-space sanitization technique (Garfinkel and Malan, 2006)