Chapter 17
Achieving Efficient Purging in Transparent per-file Secure Wiping Extensions

Wasim Ahmad Bhat
University of Kashmir, India

ABSTRACT

According to a recent Cloud Security Alliance Report, insider attacks are the third biggest threat in Cloud Security. A malicious-insider can access the low-level device, and recover the sensitive and confidential information which had been deleted by the customer with a belief that the data no more exists physically. Though proposals for secure deletion of data exist, specifically transparent per-file secure wiping extensions, however, they are not efficient and reliable. In this chapter, we propose an efficient and reliable transparent per-file-wiping filesystem extension called restfs. Instead of overwriting at file level which is found in existing wiping extensions, restfs overwrites at block level to exploit the behavior of filesystems for efficiency and reliability. We empirically evaluated the efficiency of restfs using Postmark benchmark and results indicate that restfs can save 28-98% of block overwrites which otherwise need necessarily to be performed in existing wiping extensions. In addition, it can also reduce the number of write commands issued to the disk by 88%.

INTRODUCTION

The intrusion of digital technologies into every aspect of our day to day life is continuously creating voluminous amounts of confidential and sensitive digital information which is stored in the form of directories and files. This sensitive and confidential information, which when deleted with a belief that the information has been physically erased, can be recovered even by novice users. Following are some of the incidents that are the consequence of this misbelief and had happened in recent past.

In 2004, a customer database and the current access codes to the supposedly secure intranet of one of Europe's largest financial services groups was left on a hard disk offered for sale on eBay (Leyden, 2004). In 2006, flash drives containing classified US military secrets in the form of deleted files turned up for sale in a bazaar in

DOI: 10.4018/978-1-4666-8387-7.ch017

Afghanistan (Leyden, 2006). And in 2009, the highly sensitive details of a US military missile air defense system were found on a secondhand hard drive bought on eBay (The-Daily-Mail, 2009). The situation is even worse than it seems, as the non-sanitization of storage devices continues. In 2009, a fifth study was published in an ongoing research program which was being conducted into the levels and types of information that remain on computer hard disks that have been offered for sale on the secondhand market (Jones, Valli, & Dabibi, 2009). The study revealed that over a period of five years there were clear indications that the number of disks that contain information relating to organizations and individuals is reducing. Unfortunately, it also found that because of the increasing volume of storage capacity of the disk, the quantity of non-sanitized data appears to be increasing.

Operating systems give an illusion of file deletion by just invalidating the filename and stripping it of allocated data blocks. As such, the contents of data blocks associated with a file remain there even after its deletion, unless and until these blocks get reallocated to some other file and finally get overwritten with new data. This policy is adopted as a trade-off between performance and security. Though, this allows users to recover files deleted accidentally; unfortunately, this poses a serious security threat as the files deleted intentionally can also be recovered (Rosenbaum, 2000).

In case of Cloud Service Provider, who to cut costs, conserve resources and maintain efficiency, stores more than one customer's data on a server, the hazards are magnified. As a result, more confidential but believed to be deleted data is at risk of breach via after-deletion data recovery.

There are generally two techniques to ensure secure deletion of data; 1) wiping, and 2) encryption.

Secure Deletion Using Encryption

Secure deletion using encryption can employ various encryption techniques to encrypt data before it is stored on disk and to decrypt it on its retrieval. This solution protects both deleted as well as non-deleted data. However, it suffers from several problems (C. P. Wright, Dave, & Zadok, 2003): 1) All encryption systems suffer from cumbersome and costly management of keys, 2) Encryption adds CPU overheads for most of filesystem operations, 3) Keys could be lost or broken and thus, a compromised key allows recovery of both live and deleted data, 4) Using per-file keys adds more overhead and key management costs, 5) At last, strong encryption is not allowed in some countries.

Secure Deletion Using Wiping

Secure deletion using wiping works by overwriting the meta-data and data pertaining to a file when it is deleted. In its simplest form, the filesystem or the storage media can be overwritten in its entirety and the process can be accomplished by user applications or assisted at hardware level. Unfortunately, this process is inconvenient as it erases the live data also and thus is applicable only when whole disk or filesystem sanitization is required. The most applicable and desired wiping procedure is transparent per-file wiping which can be performed at two levels of an operating system: 1) User-mode level, & 2) Filesystem level. User-mode transparent per-file wiping can be implemented by modifying the library or adding extensions to it, to support overwriting on deletion. However, this solution demands library modification, does not work with statically linked binaries, can't overwrite all the meta-data belonging to the file and can be bypassed easily. As such, it is not a feasible solution to transparent per-file wiping. In contrast, at filesystem level, all the filesystem operations can be intercepted and thus complete wiping can be guaranteed. Although many transparent per-file secure wiping filesystem extensions have been proposed, unfortunately they are not efficient (number of disk blocks to