

# Towards a Forensic-aware File System

Ryan Q. Hankins and Jigang Liu, *Member, IEEE*

Department of Information and Computer Sciences

College of Arts and Sciences

Metropolitan State University

St. Paul, MN 55106

[hankinry@go.metrostate.edu](mailto:hankinry@go.metrostate.edu) and [jigang.liu@metrostate.edu](mailto:jigang.liu@metrostate.edu)

**Abstract** *As the Internet has become an indivisible part of our daily life, the crime associated with it has increased dramatically as well. However, the dominant approaches currently used to fight against the cybercrime focus only on how to collect and preserve digital evidence after a crime has been committed, in which some of digital evidence could not be collected due to the design of file systems as well as the configuration of operating systems. In this paper, we proposed a proactive approach in dealing with cybercrime. We believe that with a forensic-aware file system and a properly configured operating system, the effectiveness and soundness of digital evidence will be significantly improved.*

**Index Terms**—File systems, Computer Forensics, Forensic-aware File System

## I. INTRODUCTION

Computer systems initially developed in relative isolation. One user at a time programmed then and input data. Over time, these systems expanded dramatically. Multitasking systems allowed several users to share a single computer system. Decreasing hardware costs, large computer networks and increasingly sophisticated software provided a number of advantages: users could more easily collaborate with each other, and computer systems could be better utilized. As computers became ubiquitous, they came into use in homes and businesses with few exceptions.

Just as the variety of applications of computer for homes and businesses increased, criminals, too, found that computer systems could be used both to aid in the commission of crimes, and to commit new crimes involving only computers. Since the use of computers in crimes did not influence the original design of computer system, early designs for computers did not include features to enforce security measures or provide evidence for investigators to recover evidence of a crime.

Automobiles and airplanes have evolved over time from fulfilling their basic function of transporting people rapidly from place to place, to containing “black boxes” that record flight and travel data. Such data can be used for several purposes, including improving vehicle safety by eliminating causes of accidents, and more effectively determining who is at fault in an accident.

Just as vehicles have benefited from the maintenance of data prior to a crash, certain computer systems can also benefit from the retention of data in anticipation that it may be useful in the event that the computer systems comes under forensic investigation. Increased retention of forensic data over time can aid in reconstruction.

Accident investigators use forensic data such as the speed, and the application of brakes and the accelerator from a vehicle to prove or disprove their hypotheses regarding the

cause of a crash. The more information that is available to them, the better they can construct the series of events that led to the crash, and ultimately determine its cause.

The most important source of information in a computer forensics investigation may be a file system. The file system contains the data stored by the users of a computer system, as well as data stored by the system itself. Currently, investigators must rely on the data available on the file system at the time of the investigation to reconstruct the string of events that occurred, ultimately leading to the investigation. They are constrained by what is currently contained in the file system to conduct the investigation.

One technique investigators use to improve temporal reconstruction, as summarized in Table 1, is the use of unallocated portions of file systems as evidence. When a file system stops using a block of data, the data remains within the block itself, but the metadata describing it indicates that the block is unallocated. Forensics investigators can use this to see what data might have been deleted from the file system. Although unallocated data is not maintained in a structured manner by the file system, it may prove useful.

However, if deleted data is useful for certain systems, how might retaining specific deleted or overwritten data improve the ability of investigators to perform forensic analysis? Certainly, this is comparable to a flight data recorder. Instead of using data from deleted portions of a file system, a new file system can record specific data and retain it for potential future forensic investigation. In this manner, what is retained and what is destroyed is specifically chosen.

AFF, *Advanced Forensic Format* [7], is one of well-known proposals in dealing with how to store digital images of hard disk or any stored digital data for potential investigations. However, AFF can only provide digital evidence that has been previously stored on a hard disk or a digital device.

Instead of a reactive approach, we propose, in this paper, a proactive approach in designing a forensic-aware file system

with a properly configured operating system to preserve critical information for potential future investigations. In the following section, related work and research are reviewed. A forensic-aware file system is proposed in section III. In section IV, we discuss a possible implementation of the design. The performance analysis is covered in section V. Finally, in section VI, conclusion and future work are presented.

## II. RELATED WORK

An important technique in forensic analysis is that of hypothesis testing, in which an investigator proposes an explanation for the state of a system, then verifies that hypothesis using the available evidence. Using such a model to create and test whether hypotheses that result in the final state of the system, an investigator can use this model as a basis for digital investigation: by applying the hypothesis-based model to any investigation, one can determine whether the technique used in the investigation has a scientific basis, allowing its use in court [5].

In addition, we discuss current means of obtaining file system forensic information [6,9], as well as other models for doing hypothesis testing, such as process labeling [2]. Used in common with these models, proactive forensics promises to emerge just as surveillance technologies have: it allows administrators to keep close tabs on their computer systems, and to provide improved information about the behaviors of users after they have used the system. In addition, it promises to provide a deterrent effect against those who would misuse systems, just as criminals avoid security cameras.

File system forensic analysis is typically performed by making a forensic image of a disk, then analyzing that disk using commercial or open source forensic tools. Commercial file system forensic tools include EnCase and the Forensic Tool Kit. Similar open source tools, such as the Sleuth Kit, are also available. Each of these tools supports a limited number of file systems, and can perform a number of operations on each file system, such as searching for files of certain types.

**Table 1 Sources of temporal forensic data**

Category	Pros	Cons	Exemplary Systems
Concurrent versioning file systems	Consistent tracking of changes to the same file	Burdensome to users No continuous data retention	CVFS
Journaling File systems	Transparent to users	Relatively easy for a sophisticated user to overwrite files upon deletion Ad hoc retention of file data	XFS ZFS EXT3
Backups	Useful on virtually any file system Potentially long-term retention of forensically useful data	No continuous data retention	EnCase Forensic ToolKit Sleuth Kit
File System Snapshots	Low overhead means of retaining file data	No continuous data retention Must be initiated by an administrator	A feature of ZFS

File systems forensic tools can extract the data currently in a file system, just as the operating system would access it. The tool understands the data and metadata that compose a file system, and allows the investigator to traverse directories, open files, and perform searches on the data. The advantage to using a forensic tool to gather information is that the tool is designed not to modify the file system. Additionally, tools may support search features that are likely to come into play in forensic analysis, such as the ability to organize data by file type; it is common in an investigation, for graphics files to be interpreted by the forensic tool; For example, forensic tools can provide the capability to view all the images on a system in one operation.

Another important aspect of file system forensic analysis is the ability of the forensic tool to examine a file system's free space. When a user deletes a file from a file system, the file system simply removes the file's entry from the file system's metadata, and returns the blocks allocated to the file to the space available for allocation in the file system. Therefore, remnants of deleted files can remain indefinitely within this

free space, depending on how soon the file system replaces the space with new data. File system forensic tools can take advantage of these remnants to reconstruct deleted files from the file system, providing potential clues to investigators.

The current state of computer system forensics is akin to security through obscurity. Parts of computer forensics are dependent on an assumption that the intruder is unaware of investigative means of recovering information about what happened to the computer system. Just as a burglar wears gloves to a crime scene in anticipation of police using fingerprints to identify him or her, as intruders become aware of computer forensics investigative techniques, they are increasingly likely to use methods to thwart that investigation [10].

File system forensics is dependent on the retention of data in the file system. Ad hoc retention results in portions of deleted files being retained in free space. However, because this portion of file system forensic analysis relies on how the file system happens to retain data, there is a great degree of

chance in whether particular data will be retained after it has been deleted.

Making it more explicit which deleted data a file system retains or destroys has the potential to improve computer forensics. Most file systems draw a distinction between two types of space: free blocks and used blocks. One of the primary purposes of the file system is to reallocate free blocks as used blocks when the operating system requests that the file system store data, and to do the opposite when the operating system deletes data. When a block of data is deleted, that block typically remains in the free blocks of the file system until the block is reallocated, at which point it is overwritten with new data. Which block a file system allocates when storing data depends on the file system implementation; file systems may optimize allocation for performance or locality, for example.

By drawing a distinction within the free space of a file system between unused space, and space that contains data that has potential value for computer forensics, it becomes possible for a file system to anticipate forensic analysis, and maintain certain data in a more structured and permanent format, than simply retaining that data until it is again allocated by the operating system.

### III. DESIGN OF A FORENSIC-AWARE FILE SYSTEM

Several sources exist for temporal forensic data. An investigator can correlate data from these sources over a timeline to aid in computer forensic reconstruction, supporting or contradicting the investigator's hypothesis. Sources for such data include data backups, concurrently versioning file systems, and file systems themselves. Additionally, some file systems, such as Sun's ZFS, allow for clones or snapshots of file systems

#### A. Data Backups

A series of backups performed on a system over a period of time can be useful to investigators by providing insight into how a system changed before an investigation took place.

Backups are used for two main purposes: they provide a secondary means of recovering data in the event the primary means of storage is damaged, and they provide a means of restoring individual files and directories for a user who accidentally deleted, modified, or damaged his or her files [6].

Due to the nature of backups, they can be useful in an investigation if the series of events the investigator is reconstructing occurred over the course of several backups.

Backups have a significant drawback, however. Because they are conducted on a periodic and not a continuous basis, and because only the in-use blocks of a file system are backed up, it is possible for an intruder to circumvent the backup system. One means of retaining data over a longer period of time, but avoiding backup of that data within a corporate environment is to utilize two separate systems, each scheduled for backup at different times. By moving the data from the system with the later backup to the system with the earlier backup between backup times, the data will not be backed up, allowing a savvy intruder to effectively eliminate the potential that incriminating data would later be available on the backup system for retrieval.

#### B. Versioning File Systems

Versioning file systems provide another means of retaining data over a period of time. Such file systems record changes to files as the changes are made; thus a series of older revisions of a particular file can be retrieved at any given time. A major difference between backups and versioning file systems is that backups are made to a separate media, and versioning file systems keep records of files on the same media as the original. Versioning file systems are similar to revision control systems, but they are transparent to the user [9].

Like backups, versioning file systems usually retain periodic revisions of files, saving storage space when only a minor change is made to a file, but leaving the possibility that some change to a file could escape the system, particularly if an intruder were knowledgeable about the details of the versioning file system.

#### C. File System Snapshots

Certain modern file systems, such as Sun's ZFS, allow instantaneous clones and snapshots of a file system. The file system does this on the request of an administrator. At the point of the snapshot, the file system marks all the data currently in the file system as part of the snapshot. Then, the system implements a copy-on-write strategy in which any data changed on disk do not update the old disk blocks, but rather are written into new blocks. The file system tracks which blocks belong to which snapshot [13]. In this way, the file system can access file system data as it was in the snapshot at any future time, but this is transparent to users. Both snapshot data and current file system data remain available, and because of the copy-on-write strategy, only blocks containing changes are duplicated, resulting in advantages in performance and space consumption.

A forensic-aware file system should adhere to several principles in its design, including reliability, transparency, completeness, compatibility and portability, configurability and extensibility, performance, and robustness.

A forensic file system should perform as reliably as a file system lacking forensic capabilities. Users should be able to perform all the operations on the virtual file system (VFS) just as they would on another file system. Similarly, transparency and completeness should allow the file system to appear to the user as any other file system would. Thus, the set of operations that could be performed on such a file system should be a superset of those available in the standard VFS.

Such a file system should be portable. Many modern file systems are supported on multiple architectures. Providing a file system that is common to the extent that several operating systems are able to use the file system after it was constructed and used on any of those operating systems provides a substantial degree of abstraction: tools to examine the file system must only be developed once, and an investigator wishing to understand the internals of the file system can use that knowledge regardless of the operating system.

The file system should be configurable and extensible. Different environments require substantially different degrees of computer forensic capabilities. Additionally, forensic data that is highly valuable in one environment may be useless in another. Providing a high degree of configurability allows a

system to gather data that is likely to be useful, and to allow experimentation at an abstract level with different strategies for data retention.

When constructing a file system that adds data retention features to the features already in other file systems, there is likely to be a performance impact, since a file system that retains additional data has the overhead of maintaining that data in addition to the data that a traditional file system maintains. It is unlikely to expect that a forensic file system's performance would match that of another file system, though a reasonable expectation for performance is to strive for performance of a forensic file system, storing both user data and forensic data, to be similar in performance to a traditional file system retaining an equivalent amount of data as user data.

Finally, a forensic file system should be robust: it should be difficult for users to circumvent the mechanisms of the file system to retain forensically useful data. Data should be retained continuously, and an administrator should be able to retrieve the data in a meaningful format.

#### IV. PROOF OF CONCEPT

To implement a prototype of a forensic file system, it is simplest to leverage existing systems to the greatest extent possible. Since most existing file systems maintain lists of data blocks that are in use, and ones that are free, allocating and deallocating disk blocks as they are needed, making modifications to a file system to make it accommodating to the retention of forensic data is, at the most basic level, a matter of adding a third list of data blocks to the two existing lists to make those lists more fully retain forensic data.

This paper describes the high-level design for a forensic-aware file system, its features, and the rationale for each feature included. The file system is designed to serve a

variety of needs, and to be configurable in such a way as to meet varying trade-offs optimally in different environments.

A forensic-aware file system is at a disadvantage to traditional file systems in several ways. The additional demands on a forensic file system likely reduce its ability to perform as fast as traditional file systems. In addition, because of the additional data such a file system retains to aid investigators, a forensic file system may have space requirements in excess of what a traditional file system would require. Table 2 shows how a forensic file system differs from a traditional one from several perspectives.

We make several basic assumptions in the implementation of such a file system. First, we assume that all activities running on the machine make some irreversible change to the disk that is later detectable. Second, we assume that an intruder is unable to reverse the changes in a disk in a way, rendering the intruder's activities invisible to forensics investigators. Third, we make the assumption that an intruder is unable to circumvent the data retention process, preventing any data describing his or her activities from being retained.

Although these changes are not considered in depth in this paper, certain methods should help to make it difficult for intruders to cover their tracks. A device on which data can be read and written, but not modified would help in this regard. This could be implemented on separate media from the system disk, or as a separate partition on the system disk that does not allow modification to data that has been written. Media that is used for data retention could be accessed in such a way as to make it impossible for an intruder to modify data once it has been written. Software or hardware constraints could provide the means to prevent an intruder from destroying incriminating data.

**Table 2. A comparison of file systems**

Affected Group	Current method of file system forensic analysis	Method of file system forensic analysis using forensic file system and tools
Users	Users can delete any data, and the recovery of deleted data depends on whether the data happens to have been preserved on the file system. Users can change a portion of a file, and only those changes will be reflected on disk; there will not necessarily be any indication that a change was made.	Users can delete data, and the recovery of deleted data depends on whether the forensic options of the file system indicated that the data should have been retained. Data retained for forensic analysis remains essentially transparent to users.
Administrators	Administrators have little control over what data is retained by the file stem once it has been deleted	Historical file system data and metadata can be saved according to a heuristic, and later retrieved, rather than based on the idiosyncrasies of the file system, and the choices of users
Intruders	An intruder who gains administrative access to a machine can modify files to attempt to hide evidence of the attack.	An intruder can modify files, but historical versions of the files will be kept, indicating the actions of the user.
Forensics Investigators	Investigators must rely on the idiosyncrasies of the file system to retrieve deleted data.	Since specific deleted data is retained or destroyed, the investigator can access exactly the data as determined by the heuristic.

We also assume that when an intruder causes a process to do something on the system, that process in some way modifies the disk in a way that can be retained. It is difficult, however, to guarantee, that any activity will cause a modification to the disk, as it is difficult to guarantee that no

way exists for a program executed on the system to be circumvented in terms of logging to disk. This is left as an area for additional investigation: How can we guarantee that a record of intrusive activities is maintained?

The file system could be prototyped in a way that would use an existing file system, in addition to storage external to the file system to retain forensic data. By monitoring a file system continuously for changes, and recording data that fit a pattern each time the file system changed, one could simulate the system.

A prototype would allow one to record changes in file system data and metadata for forensic analysis. At this point, one could perform forensic analysis on the data. This would allow a forensic investigation of the file system, showing the effectiveness of the heuristic, in addition to proving the usefulness of the file system itself.

## V. PERFORMANCE ANALYSIS

Determining whether a particular piece of data is valuable for computer forensics is an issue of foresight versus hindsight. With perfect hindsight, at the time of a forensic investigation, a forensic investigator could go back in time and view the data most relevant to the investigation. Since a file system must choose to save or discard data much earlier than the investigation occurs, it is relevant to consider how data might best be categorized in an attempt to preserve what is likely to be relevant in an investigation.

There are competing factors when designing algorithms to preserve data for forensic analysis. Consider two potential strategies of preserving and destroying data: first, preserving every change to a file or file system metadata, and second, destroying bit of information other than what is held in the in-use portion of the file system.

The first strategy is likely to consume substantial disk space, and possibly reduce performance of the system, as the file system is able to allocate from fewer blocks when it makes the decisions about how best to store incoming data. Over time, such a strategy could lead to storage requirements significantly in excess of the portions of the disk that would be used under a traditional file system. Such a strategy, however, would be most advantageous in the event of an investigation: an investigator would receive the gift of perfect hindsight, allowing him to construct a user's view of the file system at any point in time.

In contrast, the second strategy would function much as file systems do today, with the difference that data, once deleted, would be permanently overwritten from the disk, and likely unrecoverable. This would provide the opposite trade-offs from the first method: instead of providing an investigator with a perfect hindsight, the investigator would be left with only the current contents of the file system, and any examination of free space would be fruitless. Correspondingly, the file system would use the least disk space, but would prove least beneficial for forensic investigations.

Using a heuristic, it may be possible to determine which types of files are most likely to be useful in an investigation. If the heuristic can make a reasonable determination as to what data is likely to be most relevant, and what is best discarded, the trade-off between retaining data and sacrificing system resources is best resolved.

Such a forensic file system could be burdensome to computer systems. In addition to requiring additional storage

resources, the system will somewhat detrimental to performance. Although the best way to determine the actual system load might be empirical testing, we can consider how the file system might place additional demand on system resources.

### A. Storage Requirements

First, requirements for additional storage are obvious. Whenever additional data is retained, that data takes up additional space on disk. Of particular concern is data to which frequent changes are made, resulting in many successive copies of that data being stored. Using a heuristic that stores large quantities of data in this manner may result in significant additional storage requirements. This underscores the need for the development of a heuristic that attempts to store only the most important data. In addition to the heuristic, the amount of storage required is likely to be a function of how the file system is used. File systems on which data is frequently read and rewritten substantially have the potential to retain much more data than file systems that are primarily read-only.

### B. Performance Impacts

Second, the file system is likely to impact performance in several ways. Running a heuristic on file system data will require some processing time. This processing time will be required when file system data is deleted, or when file system data that has been retained is scheduled to be reevaluated. Another factor demanding processing time is the retention of data itself. Data blocks that are deleted but retained must also be retained in metadata: the file system must keep track of the blocks that were deleted. This is in contrast to a traditional file system, in which deleted blocks are simply returned to the list of free blocks in the file system, from which the file system can allocate when it needs to store additional data.

### C. Cost of destroying discarded data

A similar cost of resources is that of zeroing blocks that will not be retained. In traditional file systems, blocks that are returned to the free list are left unchanged, under the assumption that the disk is a secure part of the system, and the contents of those blocks can only be accessed from the system itself. However, with this file system, when those blocks are discarded, the block itself is filled with zeroes, so the contents can no longer be recovered. This is likely to cause some burden, particularly on systems that frequently delete, truncate or replace files.

### D. Reducing System overhead

There exist some means of reducing the costs associated with the file system. Compression and hashing have the potential to reduce redundant data within the file system to some extent: since much retained data is likely to replicate other data, these schemes may substantially reduce the amount of data that must be retained. Of course, these schemes come at the cost that when data is compressed and decompressed or hashed, that requires processor time, increasing the load on the system, trading off additional processor time for reduced storage space.

### E. Performance Impacts of Data retention

A final cost is more implicit: when modern file systems allocate blocks in which to store file data, they do it according to an algorithm that positions blocks on disk in a fashion for optimal performance. As the file system becomes increasingly full, the available blocks from which a file system can choose to place data decrease. The file system must therefore place data in a less optimal location than it would have if more optimal blocks were available. Because a forensic file system uses certain blocks for forensic data storage, it limits the locations in which the file system can store data, ultimately reducing the file system's read and write storage. Certain pathological cases may substantially impede performance, for example, if a file is truncated and appended repeatedly, retained blocks may cause the current data blocks of the file to become scattered around the disk.

### F. System Optimization

Certainly, using a forensic file system is not a means to improve system performance. Instead, as with any feature, the benefits must outweigh the disadvantages. The need to retain data for forensic purposes must outweigh the need to obtain absolutely optimal performance from the system. With that in mind, the detrimental effects of a forensic file system on performance can be mitigated. By designing a heuristic to retain only the most relevant data, using high speed storage, and using storage with enough capacity that it does not substantially burden the disks' storage capacity with forensic data, the performance impact of such a system may be minimal. Furthermore, a configurable computing system allows for the best of both worlds: administrators can tailor the system to their needs, reducing overhead or increasing data retention, choosing the trade-off that best suits the needs of their environment.

## VI. CONCLUSION AND FUTURE WORK

The design and implementation of a forensic-aware file system has the potential to become a boon to proactive computer forensics. Akin to surveillance systems in more traditional forensics, proactive systems such as the forensic-aware file system have the potential to serve as a deterrent to system abuse, by providing a disincentive to the misuse of a system: intruders are less likely to subvert a system for unacceptable purposes if they are aware that their activities are more likely to be intercepted.

The use of such a system could add real value to the forensic process. By providing investigators with reliable, temporal information indicating how a file system changed over time, it increases the reliability of the results of an investigation. While there is no silver bullet for computer forensics: investigations will continue to rely on multiple source of data for some time, increasing the forensically valuable information retained in one of the most important sources of forensic information, the file system, is a step forward in the march toward proactive computer forensics.

Using a model for forensic reconstruction, the file system proposed in this paper offers to improve the ability to conduct an accurate investigation. Such advances, however, are not without concern or cost. While the widespread use of a

forensic-aware file system might improve the reliability of investigations, ethical concerns. Moreover, due to the overhead required by a forensic-aware file system on system resources, a configurable implementation of the system seems more appealing.

Clearly, substantial work remains in the development of such an integrated forensic system. In addition to the development of the file system itself, and forensic tools to support that file system, there remains the question of how one best retains the data. Worthwhile avenues include investigations of performance of a forensic file system, and an evaluation of which data is most valuable to preserve for forensic purposes. In addition, a prototype implementation of the file system would enable us to show how computer forensic analysis would improve.

## REFERENCES

- [1] Boyd, C., and Forster, P. Time and date issues in forensic computing—a case study. *Digital Investigation 1*, 1 (February 2004), 18–23.
- [2] Buchholz, F. Pervasive Binding of Labels to System Processes. PhD thesis, Purdue University, 2005.
- [3] Buchholz, F., and Spafford, E. On the role of file system metadata in digital forensics. *Journal of Digital Investigation 1*, 4 (2004), 298–309.
- [4] Carrier, B. Defining digital forensic examination and analysis tools using abstraction layers. *International Journal of Digital Evidence 1*, 4 (2003).
- [5] Carrier, B. D. A Hypothesis-Based Approach to Digital Forensic Investigations. PhD thesis, Purdue University, 2006.
- [6] Chervenak, A. L., Vellanki, V., and Kurmas, Z. A survey of backup techniques. In *Proceedings of the Joint NASA and IEEE Mass Storage Conference (Mfarch 1998)*.
- [7] Grafinkel, S., AFF: A New Format for Storing Hard Drive Images, Communications of the ACM, Feb. 2006, Vol. 49, No. 2, pp85 - 87
- [8] <http://www-128.ibm.com/developerworks/linux/library/1-inotify.html>
- [9] Muniswamy-Reddy, K.-K., Wright, C. P., Himmer, A., and Zadok, E. A versatile and user-oriented versioning file system. In *Proceedings of the Third USENIX Conference on File and Storage Technologies (FAST) (2004)*.
- [10] Newsham, T., Palmer, C., Stamos, A., and Burns, J. Breaking forensics software: Weaknesses in critical evidence collection. In *Proceedings of the 2007 Black Hat Conference (2007)*.
- [11] Skupsky, D. Establishing retention periods for electronic records. *Records Management Quarterly (2004)*.
- [12] <http://docs.sun.com/app/docs/doc/819-5461/6n7ht6qsb?a=view>
- [13] Patzakis, J. New accounting reform laws push for technology-based document retention practices. *International Journal of Digital Evidence 2*, 1 (2003), 1–8.