January 19, 2005

Deleted File Recovery Tool Specification

Draft for SC Review of Version 1.0

National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

CONTENTS

1 Preface

There is a critical need in the law enforcement community to ensure the reliability of computer forensic tools. A capability is required to ensure that forensic software tools consistently produce accurate and objective results. The goal of the Computer Forensic Tool Testing (CFTT) project at the National Institute of Standards and Technology (NIST) is to establish a methodology for testing computer forensic software tools by development of general tool specifications, test procedures, test criteria, test sets, and test hardware. The results provide the information necessary for toolmakers to improve tools, for users to make informed choices about acquiring and using computer forensics tools, and for interested parties to understand the tools capabilities. Our approach for testing computer forensic tools is based on well-recognized international methodologies for conformance testing and quality testing. This project is further described at http://www.cftt.nist.gov/.

The CFTT is a joint project of the National Institute of Justice, the research and development organization of the U.S. Department of Justice; the National Institute of Standards and Technology Office of Law Enforcement Standards and Information Technology Laboratory, and other agencies, such as the Technical Support Working Group and the IRS. The entire computer forensics community participates in the development of the specifications and test methods by commenting on drafts as they are published on the CFTT website.

2 Introduction

Frequently during a forensic examination, data is discovered on the target media that is not part of any active or visible file. Although this data can still be examined (e.g. string searching), as would be done for unallocated space, if the data associated with a particular file could be identified and recovered in its original form, this could provide additional useful information. An example of this would be where a graphics file, if undeleted and recovered, could be viewed—potentially providing more information than a simple string search. Many of the forensic tools used by investigators identify files that have been deleted, and allow the operator to undelete them. This may allow the investigator to examine the file in the original format (e.g. a graphics file viewer), or identify when a particular file was deleted and its original location.

To reconstruct files that have been deleted within a forensic setting, three fundamental problems have to be addressed by a deleted file recovery (DFR) tool. First, the files that have been deleted have to be identified and located. Although this could be as simple as scanning directory entries for a particular key (e.g. ‘0xE5’ in Fat 32) it may be a more complex process. This process is paramount for any recovery tool to work correctly, for if files are not correctly identified and located, they will not be part of the recovery process.

The second problem, from a file system perspective, is that the data to be recovered is *latent*, and needs the assistance of a tool to recover the data in a form that is usable by the investigator. As with most other latent data recovery, since the results depend on the output of a particular tool, it must be shown to operate correctly (i.e. undelete files correctly).

The third and final fundamental problem that needs to be addressed by deleted file recovery tools is the potential uncertainty in any recovery effort. A common problem with all residual data recovery is that since residual data is no longer maintained by the file system, there is a reduced level of confidence in the information recovered. Specifically with deleted file recovery, the data recovered may be commingled with data from other deleted files, allocated files, or even from non allocated space.

3 Purpose

This document defines the functional requirements and specifications for deleted file recovery tools used within forensics investigations.

These requirements were developed through a combination of processes including but not limited to deleted file recovery research, personal interviews with forensic investigators, and working with a focus group of individuals who are experts in the field of forensic investigation and depend on the results of deleted file recovery tools. Additionally, as this document evolves, feedback will be incorporated from a variety of sources, and will be posted to our web site at http://www.cftt.nist.gov for comments.

It is important to note that this document is limited to the defining of functional specifications and requirements for deleted file recovery tools and processes. Additional documents used in testing such as the assertions, test cases, and actual testing results will be developed.

4 Scope

The scope of this specification and requirements document is limited to software that identifies and recovers deleted files. The tools examined will be limited to those that are currently used within the general computer forensic community, as indicated from research and feedback from various focus groups. The proper or improper use of a tool is not within the scope of this specification.

The specifications and requirements for deleted file recovery are high-level, and are based on the following assumptions.

General:

- The deleted file recovery tools are used in a forensically sound environment.
- The individuals using these tools adhere to forensic principles, and have control over the environment in which the tools are used.

Tool Functions:

- Only deleted file recovery tools and functions are examined.
- Other types of latent data recovery are not part of this specification.

Tool Environment:

- Only the file systems supported by a given tool are tested.
- Each file system tested is correctly configured, and is accessible if mounted on the appropriate and receptive operating system.
- Only commonly used file systems will be part of the testing parameters. Encrypted, compressed, and distributed file systems are outside the scope of this document.

Deleted File State:

- It is assumed that the files used to test the deleted file recovery process were created and deleted in a process similar to how an end-user would create and delete files.
- Files that were specifically corrupted, modified, or otherwise manipulated to appear deleted are outside of the scope of this document.

5 Background

This section provides the technical background needed to discuss deleted file recovery tools and functions. The first section outlines a brief high-level model of a file system. Section two covers the two most common properties of file systems, which are the basis for most deleted file recovery efforts. Section three outlines some of the reference material used to understand file systems, and how various implementations may affect deleted file recovery.

5.1 Abstract Model of a File System

A file system is used to store data for access by a computer. The data is normally stored within a tree-like structured hierarchy of directories and files. File system *metadata* contains information to describe and locate every file within a given file system. Some *metadata* resides in directory entries, but additional *metadata* may reside in special files (e.g. NTFS $MFT) or other locations (e.g. UNIX i-nodes).

When a file or directory is deleted, normally the associated *metadata* entry is flagged as being no longer active. However, in most file systems, neither the metadata associated with the file nor the actual content is completely removed. This creates a situation where there is *residual metadata* (metadata remaining after a delete has occurred) that may still be accessible. However, depending on the original format and structure of the metadata, not all of it may be reachable. This would be the case for a fragmented directory, where the first data block of directory entries would be reachable even after deletion, but the remaining data blocks of directory entries are not.

5.2 File System Properties

File systems are designed to allow an operating system to have access to secondary storage in a manner that is both efficient and timely, as in the past, storage devices have been expensive, and slow (when compared to Random Access Memory). Accessing the hard drive efficiently, although implemented differently in each file system, tends to have some side effects that can be exploited to recover deleted files. Two of the most key properties are contiguous writes, and the conservative nature of file system activity.

File systems use contiguous writes if possible: Most operating systems write data to the drive in a contiguous set of data blocks or sectors if available. A given data file, provided it is not modified after being written to the disk, tends to have all the data in sequentially accessible sectors. This speeds up both the write and read processes, since the heads on the drive do not need to move to different areas on the disk to write or read data. This plays a role in data recovery, in that data from a given file, even deleted, has a high likelihood of being grouped together on the disk in contiguous data blocks.

File systems are conservative: This characteristic implies that, in order to be as fast and efficient as possible, file systems perform many activities with a minimum of changes or overhead. In the case of file deletion, in most situations, only a *logical deletion* is performed—meaning that the actual data is not erased, but the metadata that indexes the information is changed, flagged or removed. By using this technique, a file's content, no matter how large, can be "deleted" by simply modifying or removing entries in a disk index. The simplest example of this is how a windows FAT 32 file system deletes files. It locates the directory entry of the file to be deleted, and changes the beginning character in the file name to a '0xE5' hex value, and then zeros the file allocation table. This indicates to the file system that a file has been deleted, and is no longer accessible (or maintained) by the file system—yet most of the metadata and the entire file content remain.

For the most part, these common attributes assist in the recovery of data on the drive, regardless of the type of file system the data resides on. Many tools leverage the residual metadata in locating the potential file system objects, and then recover the largest amount of contiguous data.

5.3 References (Informative)

Documents and research that were of particular relevance to the deleted file recovery, background information, and the specifications and requirements document. It is important to note that these references were primarily informative.

Carrier, (2003). “File System Analysis Techniques: Sleuth Kit Reference Document.” Available at http://www.sleuthkit.org/sleuthkit/docs/ref_fs.html.

Crane, (1999). “Linux Ext2fs Undeletion mini-HOWTO.” Available at http://www.tldp.org/HOWTO/Ext2fs-Undeletion.html.

Erdelsky, (1993). “A Description of the DOS File System.” Available at http://www.alumni.caltech.edu/~pje/dosfiles.html.

Himmer, (2000). “File Systems HOWTO.” Available at http://www.faqs.org/docs/Linux-HOWTO/Filesystems-HOWTO.html.

Microsoft, (2004). “Description of the FAT32 File System.” Available at http://support.microsoft.com/default.aspx?scid=http://support.microsoft.com:80/support/kb/articles/q154/9/97.asp&NoWebContent=1.

NIST, (2004). “General Test Methodology for Computer Forensic Tools,” Available at http://www.cftt.nist.gov/

6 Definitions

Included here are definitions of terms used in this specification document. Although there may be commonly accepted definitions for some of the terms, the context in which they are applied may change their meaning.

Data Block: File system specific data allocation unit (block), usually 512 bytes or a multiple of it. Some file systems may use other terms to describe a *data block* such as Sector (in FAT file systems).

Deleted Block Pool (DBP): A conceptual collection of *data blocks* that were originally part of an FS-Object, subsequently deleted, and have not been reallocated or reused.

Documentation: The collection of materials available to the operator of a given undeletion tool or function that describes its usage, purpose, operation, or system requirements.

Estimated Content: A tool *Estimates Content* if it attempts to recover the content of a deleted file, beyond what is explicitly identified in the *residual metadata*.

File System Object (FS-Object): The fundament objects to store and organize information within a file system. The most common examples of *FS-Objects* would be Files and Directories.

Logical Order: The content of a *FS-Object* as it would be sequentially accessed.

Logical Deletion: When an *FS-Object* is deleted through metadata manipulation, without the actual object data being erased. For example, in FAT32, when an object is deleted, the directory entry is flagged, and the file allocation entries are cleared—the actual file data is not removed or erased.

Metadata: The associated periphery information or attributes that describe a FS-Object such as name, time-based metadata (creation, modification, and last accessed times), access rights, ownership, and location.

Recovered Object (RO): The object constructed by a Deleted File Recovery Tool through examining residual metadata. Due to the potential for corruption inherent with data that is no longer maintained by a file system, the *RO* and associated attributes may not completely match the original *FS-Object*. However, the *RO* is a sequence of *data blocks* with the following properties:

1. Each RO shall contain all data blocks identified from the *residual metadata*.
2. Each RO shall consist of only data blocks from the *Deleted Block Pool*.

Residual Metadata: The metadata that remains after a *FS-Object* has been deleted. In some cases there may exist more residual metadata than can be accessed. For example, if a directory is fragmented, when it is deleted, usually only the first *data block* of *metadata* is accessible, while the remaining fragmented directory information is not.

7 Requirements

The requirements section is divided into two parts. The first, *Requirements for Mandatory Features*, are those features that are critical to the operation of the given tool, and must be present. The second part is the *Requirements for Optional Features*. These features, on the condition they are present, will be used to report on the tool capabilities. If a feature is not present, then requirements for those features will not be tested.

7.1 Requirements for Mandatory Features

All deleted file recovery tools must support the following requirements.

DFR-RM-01 The tool shall support recovery efforts on file systems identified by the *Documentation*.

DFR-RM-02 The tool shall identify all deleted *File System-Object* entries in *residual metadata*.

DFR-RM-03 The tool shall report errors in constructing a *Recovered Object*.

DFR-RM-04 The tool shall construct a *Recovered Object* for each deleted *File System-Object* entry in *residual metadata*.

DFR-RM-05 Each *Recovered Object* shall include all non-allocated *data blocks* identified in a *residual metadata* entry.

DFR-RM-06 Each *Recovered Object* shall consist only of *data blocks* from the *Deleted Block Pool*.

7.2 Requirements for Optional Features

The following define conditional requirements for optional features. The requirements below are used to report on the tool capabilities. If the tool does not provide the capability defined, then the requirement will not apply.

If the residual metadata for deleted files in a given file system does not identify all file allocation units in the deleted file, the DRF tool may optionally create a recovered object that estimates the likely content of an original file identified in the residual metadata by extrapolation from drive content. This is referred to as *Estimates Content*.

DFR-RO-01: The tool shall report *Recovered Object* attributes that are recoverable from *residual metadata*.

DFR-RO-02: If the tool *Estimates Content* then each recovered *data block* shall be assigned to a *Recovered Object* no more than once.

DFR-RO-03: If the tool *Estimates Content* then the *Recovered Object* shall consist only of *data blocks* from the original *File System-Object* identified in the *residual metadata*.

DFR-RO-04: If the tool *Estimates Content* then any data blocks in the *Recovered Object* shall be in the same *logical order* as in the original *File System-Object* identified in the *residual metadata*.

DFR-RO-05: If the tool *Estimates Content* then the *Recovered Object* shall consist of the some number of blocks as the original *File System-Object*.