**Bowling Green State University**
**Computer Science Department**

# Deleted File Recovery for the Ext4 file system

Joshua Nodler  Sai Bharadwaj  S. Roy
*jnodler@bgsu.edu*  *ssaibha@bgsu.edu*  *sanroy@bgsu.edu*

## 1. Introduction

Ext4 is a commonly used file system in Linux distributions. Deleted File Recovery (DFR) plays a crucial role in tasks such as finding lost data and searching for important digital forensics artifacts. In the context of DFR, Ext4 lacks research and documentation compared to the other commonly used file systems (such as NTFS or FAT). There seems to be only one public tool named ext4magic, which is no longer maintained, that attempts to retrieve deleted files in Ext4. To improve the state-of-the-art, our current work implements a DFR tool for Ext4, called *tsk-dfr* that leverages three components of The Sleuth Kit (TSK) as building blocks. We evaluate *tsk-dfr* on multiple test images and compare its efficacy with other tools.

## 2. Background

Our research focuses on metadata-based DFR. Before changes are made to blocks within the file system (in the event of file creation, modification, or deletion), the file system logs some documentation of the changes. There is an area of the file system called the *journal* where Ext4 keeps this documentation. Compared to older versions (such as Ext2/Ext3) [1], Ext4 contains new distinguishing features. The inode of a deleted file in Ext4 no longer points to the data blocks (where the data sits), but rather are filled with zeros [2]. Another new feature of Ext4 is something called *extent*, which is part of an inode and it is to book-keep information about the file fragmentation. Extents are a mechanism to decrease the amount of space needed for metadata and are recorded in the journal. They have a basic structure that includes key information such as the size and starting address of the particular file.

## 3. *tsk-dfr* tool

The *tsk-dfr* tool attempts to dig out a copy of the inode of the deleted file from the journal. We implemented the *tsk-dfr* tool as a shell script that leverages *fls*, *jls*, and *jcat* tool of TSK, and the stages of the *tsk-dfr* tool are illustrated in Figure 1. Given an image of a Ext4 file system (FS), the *fls* tool is used to gather the list of the inodes of the files, including the deleted files. Once a target inode is selected, the FS block number is calculated. The *jls* tool takes in the image and returns the list of all journal entries. These entries in combination with the FS block number are used to determine the target journal entry y. Both the image and target journal entry number y are fed into the *jcat* tool which returns a journal block that hosts a copy of the target inode x. From here *tsk-dfr* extracts the location of the data blocks of the target file and then attempts to recover the file.

## 4. Experimental Evaluation

Along with the challenge of inodes no longer pointing to the data blocks of a deleted file, additional challenges stem from fragmented or (partially) overwritten files. We created test case file systems that aim to model practical examples and impediments to DFR. All test cases were Ext4 file systems of size 5MB. Text files were created on the file system and then deleted so that DFR was possible. Depending on the test case, files were created so that part of the data was overwritten and/or the original data was purposely fragmented. Table 1 reflects the experimental evaluation of the DFR tools on the four test cases. Two green check marks ✓✓ reflect full recovery of the file, one green check mark ✓ reflects only partial recovery, and the red X reflects the inability to recover the file [3]. Our test images represent the following scenarios: **Test Image 1** contains a deleted file that was neither fragmented nor overwritten, **Test Image 2** contains a deleted file that was fragmented, **Test Image 3** contains a deleted file that was (partially) overwritten, and **Test Image 4** contains a deleted file that was both fragmented and (partially) overwritten. The tools evaluated include ext4magic, *tsk-dfr*, and a proprietary digital forensics tool suite named Magnet Axiom. Note that the current version of *tsk-dfr* does not account for *extent trees* in Ext4 (occurring when many files are added to the file system) and our future work will aim to address that limitation.

**Table 1:** Evaluating the DFR tools on the test images

| | **ext4magic** | ***tsk-dfr*** | **Magnet Axiom** |
|---|---|---|---|
| Test Image 1 | ✓✓ | ✓✓ | X |
| Test Image 2 | ✓ | ✓ | X |
| Test Image 3 | X | ✓ | X |
| Test Image 4 | ✓ | ✓ | X |

### References

[1] Brian Carrier. *File System Forensic Analysis*. pub-AW, 2005.

[2] Kevin D Fairbanks. An analysis of Ext4 for digital forensics. *Digital Investigation*, 9:S118–S130, 2012.

[3] S. Sirivaram and S. Roy. Deleted file recovery for the Linux file system (Ext4). In *12th International Symposium on Digital Forensics and Security*, 2024.

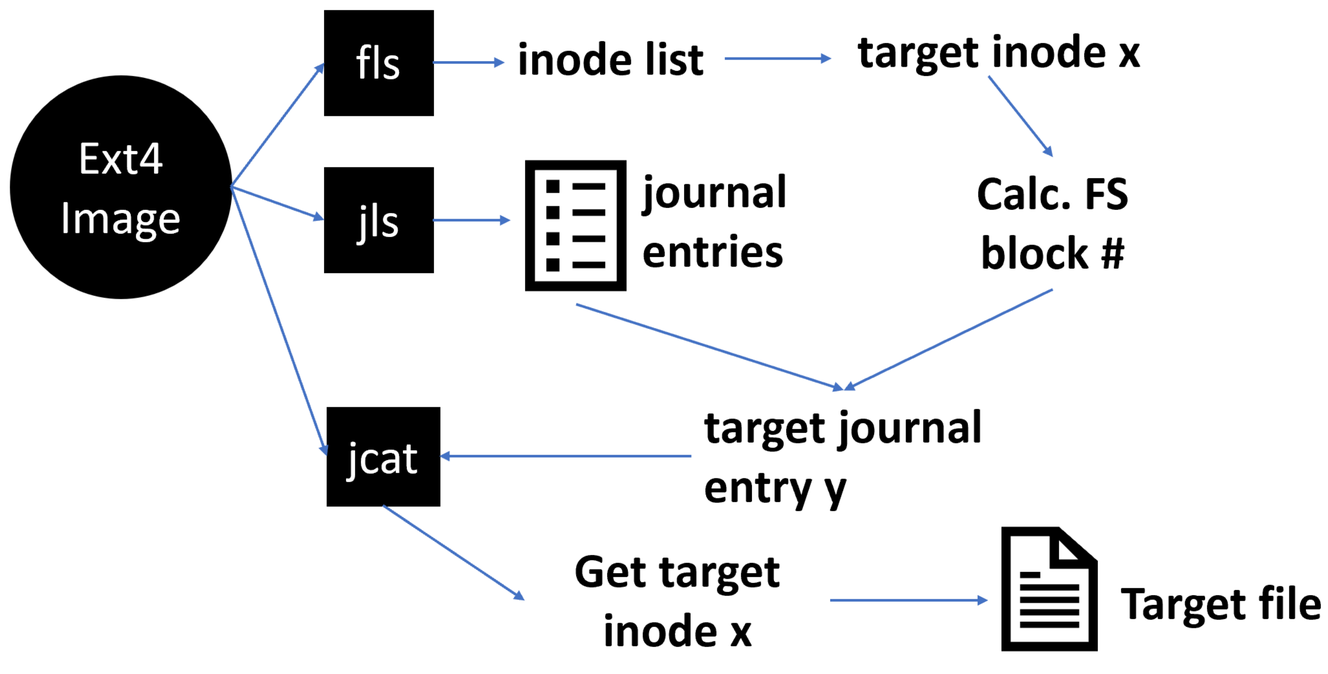


**Figure 1:** Illustrating stages in the *tsk-dfr* tool