Linux Incident Response - File Deletion in EXT4

Taz Wake | Halkyn Consulting Ltd | Originally published 2023-11-15

Overview

The EXT4 filesystem, an evolution of the earlier EXT3, is a prevalent filesystem in Linux environments. Understanding what happens when a file is deleted in EXT4 requires diving into the filesystem's architecture and the interplay of various system components. This article explores the underlying mechanics of file deletion, shedding light on the intricate processes within the EXT4 filesystem.

EXT4 Filesystem Structure

EXT4, an abbreviation for the Fourth Extended Filesystem, is designed to store and manage data efficiently. It organizes data into blocks, inodes, and directories. A file in EXT4 is represented by an inode, which contains metadata about the file but not its name or content. The actual file data is stored in data blocks, while the directory entry (dentry) links the file name to its inode.

File Deletion Process

Inode and Data Block Management

When a file is deleted in EXT4, the system performs several operations. The deletion process starts with the removal of the directory entry that links the file name to its inode. Without this linkage, the file becomes inaccessible to users.

Update of Inode Tables

Following the removal of the directory entry, the inode table is updated. The inode of the deleted file is marked as free, and its metadata (like file size, timestamps, etc.) is erased. However, the inode itself is not immediately wiped out; it remains in a reusable state.

Data Block Release

The data blocks associated with the file are also marked as free. However, the data within these blocks is not immediately overwritten. The blocks enter a pool of available blocks and may later be allocated to new files, at which point the original data gets overwritten.

Filesystem Journaling in EXT4

One critical feature of EXT4 is its journaling capability, which ensures filesystem integrity. As you can imagine, the journal plays a part in file deletion and also any attempts to recover files.

When a file is deleted in an EXT4 filesystem, key information is recorded in the journal. This data ensures the filesystem can be restored to a consistent state in case of an interruption during the deletion process. The journaling of file deletion typically involves the following types of data:

1. Metadata Updates:

- **Inode Information**: Before deleting a file, the system updates the inode, marking it as free. This change in inode status is logged in the journal. The inode entry includes metadata like file size, ownership, permissions, and timestamps. After deletion, while the inode is marked as free, these details are usually still intact until reused or overwritten, and this change is recorded in the journal.
- **Directory Entry Modification**: The deletion of a file involves removing its entry from the directory it resides in. This modification to the directory's data structure is recorded in the journal. It includes the removal of the reference (link) between the file name and its inode.

2. Block Allocation Changes:

- **Data Blocks**: The data blocks that were allocated to the file are marked as free. Information about these blocks, such as their numbers and the fact that they are now unallocated, is logged in the journal. However, the actual content of these blocks is not immediately erased and not typically recorded in the journal.

3. Transaction Information:

- **Start and End of Transaction**: The journal records the start and end of the deletion transaction. This helps in identifying complete transactions and aids in the recovery process in case of a partial write due to a system crash or power failure.
- **Sequence Numbers**: Each journal entry is tagged with sequence numbers, which help in maintaining the order of transactions and ensuring the consistency of the filesystem during recovery processes.

4. Journal Checkpointing:

- After the deletion operation is successfully completed and the filesystem state is consistent, the journal is updated to reflect that the changes have been committed. This process, known as checkpointing, marks the transaction as complete and the journal entries related to it can be flushed in subsequent operations.

Impact on File Recovery

The data recorded in the journal during file deletion is crucial for filesystem integrity but does not typically include the actual file content. This means that while the journal helps in maintaining structural integrity, the recovery of the file content after deletion relies more on the fact that the actual data blocks are not immediately overwritten and less on the journal content.

In summary, the EXT4 journal records critical changes to filesystem metadata and block allocation status during file deletion. This information is pivotal for maintaining filesystem consistency and aids in recovery scenarios, although it's important to note that the journal does not store the actual content of the deleted files.

The Role of System Calls

Syscall for File Deletion

The deletion of a file is initiated by a system call, typically unlink() or unlinkat(). These calls inform the operating system to remove the directory entry associated with the file.

Interaction with the VFS

The system calls interact with the Virtual File System (VFS) layer of the Linux kernel, which abstracts the filesystem specifics and provides a uniform interface for file operations.

The Impact of Buffer Cache

Linux utilizes a buffer cache to optimize disk operations. When a file is deleted, the changes (like inode and data block updates) may first be made in the cache before being written to disk. This process can lead to a delay between the deletion command and the actual disk write operation.

Deletion vs. Truncation

Truncation (truncate() syscall) is a process similar to deletion but differs in that it only removes the content of a file while keeping the inode and directory entry intact. Understanding this distinction is crucial for filesystem forensics and data recovery.

Recovery Possibilities

Since EXT4 does not immediately overwrite inode and data block information, there is a window for data recovery tools to salvage deleted files. These tools scan for inodes marked as free but still containing pointers to data blocks with intact data.

Summary

File deletion in EXT4 is a multi-step process involving directory entries, inodes, data blocks, system calls, and journaling. While deletion makes a file inaccessible, remnants of data persist until overwritten, offering a window for data recovery. This intricate process highlights the complexity and efficiency of the EXT4 filesystem in managing data.

Understanding these mechanics is essential for professionals dealing with filesystem management, data recovery, and system forensics in Linux environments. The architecture of EXT4 not only optimizes data handling but also provides mechanisms for data integrity and recovery.

If you want to know more about Linux forensics, have a look at the SANS Institute FOR577 (Linux Incident Response) course: https://sans.org/for577

About Halkyn Consulting

Halkyn Consulting Ltd is a UK-based security and risk management consultancy providing 24/7 incident response, digital forensics, threat hunting, virtual security leadership (vCISO) and security compliance support to organisations worldwide. We deliver clear, practical, vendor-independent advice based on real-world threats — no hype, no upsell.

More free security resources are available at www.halkynconsulting.co.uk/security-resources.

Web: www.halkynconsulting.co.uk | **Email:** info@halkynconsulting.co.uk | **Tel:** +44 1244 940858