File Deletion and Recovery in XFS File Systems

Taz Wake | Halkyn Consulting Ltd | Originally published 2023-11-21

Overview of XFS

XFS, a high-performance 64-bit journaling file system created by Silicon Graphics, Inc. (SGI) in 1993, is notable for its efficient execution of parallel input/output (I/O) operations. It is designed with allocation groups, which are subdivisions of physical volumes, enabling scalability of I/O threads, file system bandwidth, and file size across multiple storage devices. XFS is the default file system for certain Linux distributions like Red Hat Enterprise Linux. It employs metadata journaling and write barriers to ensure data consistency and uses extents within B+ trees for space allocation, significantly improving performance, particularly for large files.

The File Deletion Process in XFS

1. **User-Level Command Execution**: The **rm** command, input by the user, begins the file deletion process. This command interfaces with the file system to initiate the removal of a specified file.

2. **System Call Translation**: The **rm** command is converted into the **unlink()** system call. This system call instructs the operating system to remove the link between the file name and its inode (the metadata container for the file), effectively removing the file's entry from the directory.

3. **Immediate Metadata Update**: XFS's journaling feature ensures that updates, such as file deletions, are first recorded in a serial journal before the actual disk blocks are updated. This journaling process, which involves a circular buffer of disk blocks, is crucial for maintaining data consistency, particularly in the event of a power outage or system crash.

4. **Allocation Groups and Extent-based Management**: In XFS, files and directories are managed within equally sized linear regions known as allocation groups. Each group manages its inodes and free space independently, allowing for parallelism in I/O operations. This structure is particularly advantageous for systems with multiple processors or cores. When a file is deleted, its data blocks, managed using variable length extents (contiguous block sequences), are marked as available in their respective allocation groups. This process is facilitated by a pair of B+ trees for each allocation group, which are indexed by the length and starting block of free extents, making the allocation of these extents highly efficient.

5. **Delayed Allocation Technique**: XFS uses a lazy evaluation technique known as delayed allocation. When a file is written to the buffer cache, XFS reserves the appropriate number of file system blocks but does not immediately allocate them. This

allocation only occurs when the data is flushed to disk, which increases the likelihood of the file being written in a contiguous block group. This technique reduces fragmentation and enhances performance. During file deletion, the delayed allocation feature plays a role in how and when the space previously occupied by the file is reused.

GUI-Based Deletion: A Comparative Analysis

Deleting files through a GUI introduces additional layers of complexity:

- **User Interaction**: When a file is moved to the Trash, it's typically relocated to a hidden directory, preserving its data and metadata.
- **Trash vs. rm**: Files in the Trash can be restored, offering a safety net against accidental deletions. In contrast, **rm** bypasses this stage, directly invoking the deletion process without the intermediate step of moving to Trash.
- **Final Deletion from Trash**: Emptying the Trash or deleting a file from it triggers a process akin to **rm**, permanently removing the file's directory entry and updating the inode table.

File Recovery Mechanisms

The potential for file recovery in XFS is primarily due to the delayed overwriting of data blocks:

- **Metadata and Carving Techniques**: Recovery tools may analyse file system metadata to gather information about recently deleted files. Additionally, data carving methods, which involve scouring unallocated blocks for file signatures, are employed to piece together recoverable data.
- **Scalpel and Foremost Mechanics**: These tools operate by identifying specific file headers and footers within the unallocated space, and reconstructing files from these fragments. They are configurable to recognise various file types based on unique headers and footers, enabling the recovery of a wide range of data types.

Summary

The deletion and recovery processes in XFS are multifaceted, involving a combination of user-level commands, system calls, journaling, allocation groups, extent-based management, and delayed allocation. Understanding these mechanisms is crucial for effective data management and recovery strategies.

You can find out more about file recovery in Linux and have the chance to analyse evidence collected during a realistic intrusion, on the SANS Institute course **FOR577 Linux Incident Response and Threat Hunting**. Find out more at 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