Filesystems Security and Reliability: A Forensic Proof of Concept

Comparing FAT32, NTFS, and ext4 in Data Recovery and Tamper Detection

This presentation examines the security and reliability features of three common filesystems through a forensic lens. Our team conducted a proof of concept study to evaluate how these filesystems perform under various scenarios including corruption, deletion, and metadata tampering.

By PADONOU Julio and Etienne FUH

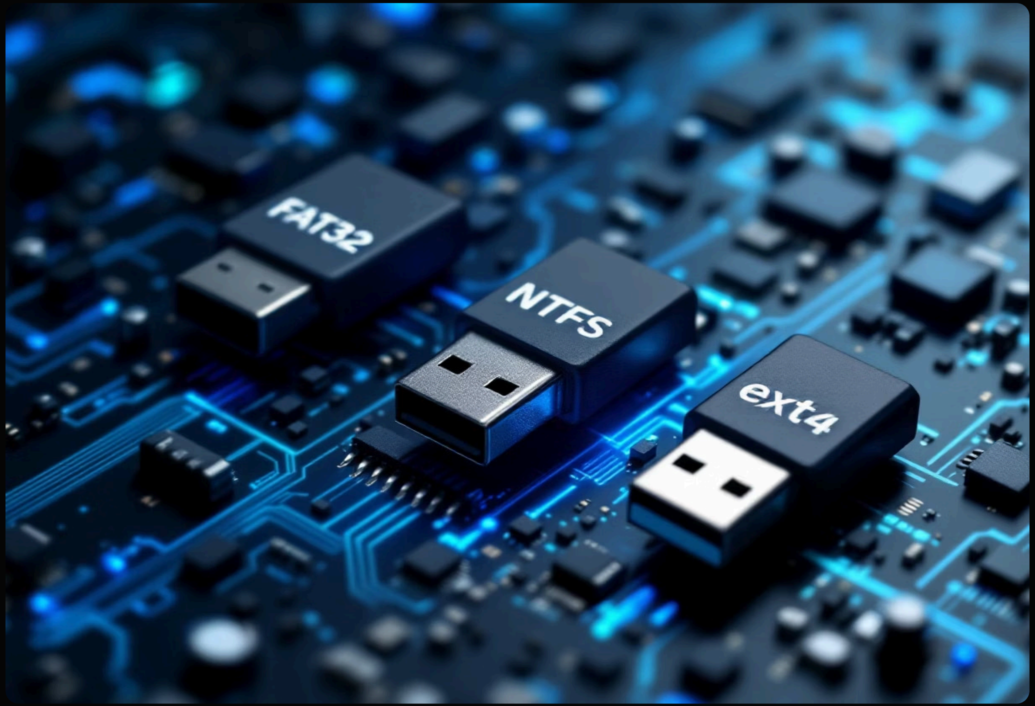
J **par Julio Padonou**

Why We Studied Filesystems

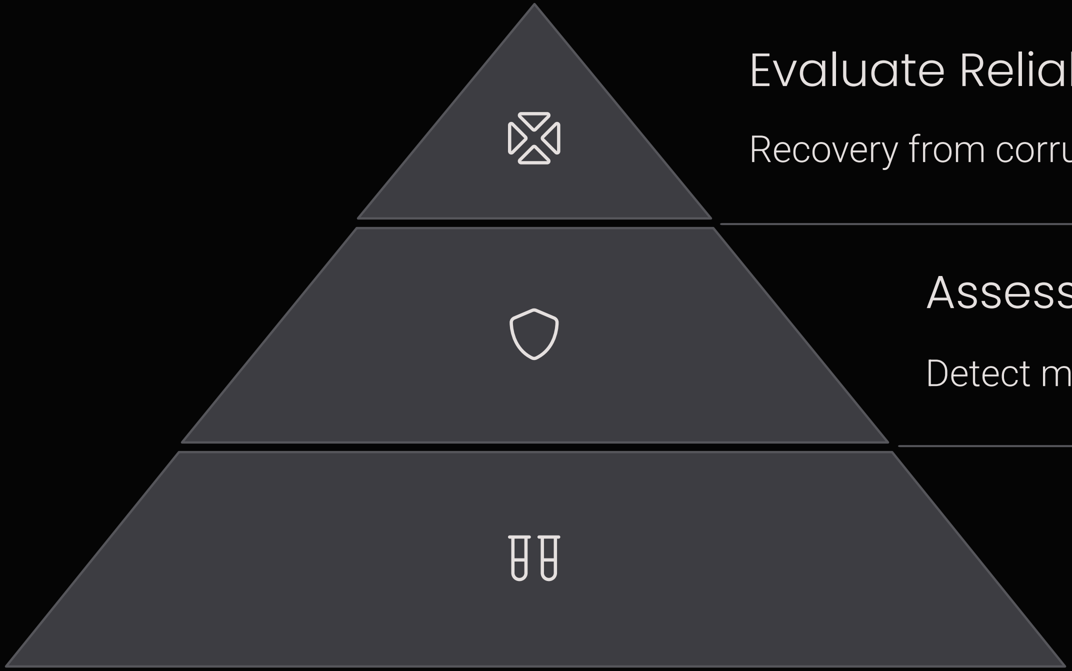
Our goal was to compare the security and reliability of FAT32, NTFS, and ext4 in a forensic context. We selected these filesystems for their distinct strengths:

- FAT32: Simplicity and broad compatibility
- NTFS: Robustness, journaling, and security features
- ext4: High-performance journaling, Linux-native

For our testing environment, we used Ubuntu 22.04 LTS VM and dedicated USB drives. We expected NTFS and ext4 to excel in our tests, while FAT32 would likely lag behind.



Our Goals and Expectations



Evaluate Reliability

Recovery from corruption/deletion

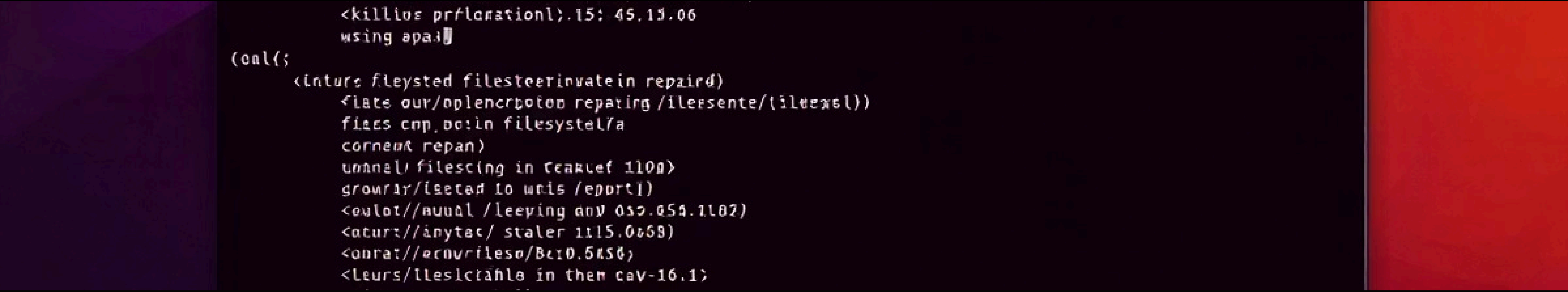
Assess Security

Detect metadata tampering

Test Three Scenarios

Corruption, Deletion, Metadata Attack

We anticipated that ext4 would lead in performance due to its advanced journaling capabilities. NTFS would likely show strong recovery and security features, while FAT32 would struggle without journaling support.



```
            <killius prflomationl}.15: 45.13.06
            wsing apa.i
(oal(;
      <inturs fleysted filestoerinvatein repaird)
            <iate our/oplencrboton repating /ilersente/(ilexsl))
            fiecs cnp.oo:in filesystal/a
            cornemt repan)
            unnnal/ filescing in ceaue£ 110g)
            growr1r/isetad to wnis /eport])
            <oulot//auual /leeying any 0s3.054.1102)
            <oturt//anytac/ staler 1115.0069)
            <onrat//ecnvrileso/BcrD.5K56)
            <leurs/lleslcianla in them cav-16.1)
```

How We Tested the Filesystems

Environment Setup

Ubuntu 22.04 LTS VM with three USB drives formatted as FAT32, NTFS, and ext4 respectively

Scenario 1: Corruption

Corrupted corruption_test.txt with dd by overwriting 10 bytes

Scenario 2: Deletion

Deleted test1.txt using the rm command

Scenario 3: Metadata Attack

Altered test2.txt's modification time (mtime) to 2026-01-01

We employed various tools including fsck.vfat, ntfsfix, fsck.ext4, testdisk, and hexdump to analyze and attempt recovery in each scenario.

Scenario 1: Data Corruption

Filesystem	Tool Used	Result
FAT32	fsck.vfat	Repaired structure, file and metadata still corrupt
NTFS	ntfsfix	Fixed MFT, file and metadata still corrupt
ext4	fsck.ext4	Repaired, file and metadata still corrupted

We simulated corruption by overwriting 10 bytes of corruption_test.txt using the dd command. Surprisingly, none of the filesystems were able to fully recover the corrupted data. This contradicted our expectation that ext4 would perform better in this scenario.

The results highlight an important limitation: filesystem repair tools focus primarily on structural integrity rather than data recovery.

Scenario 2: File Deletion

Original File

test1.txt

Deletion

Removed with rm command

Recovery Attempt

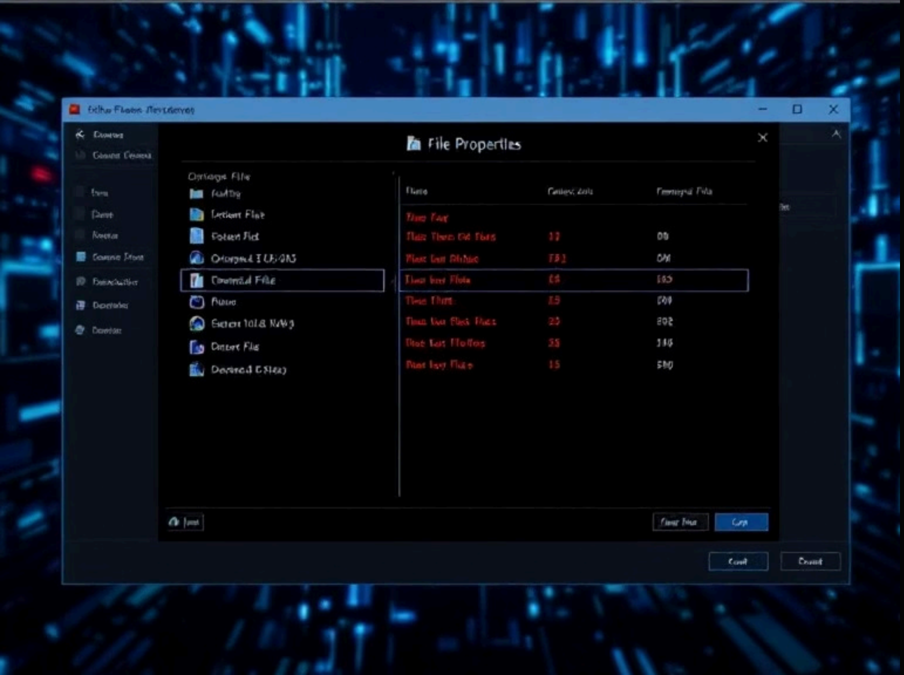
Using TestDisk tool

Results Analysis

Comparing recovery success

In our deletion tests, NTFS showed the strongest recovery capabilities, likely due to its Master File Table (MFT) design. The ext4 filesystem provided partial recovery through its journaling mechanism, while FAT32 struggled with recovery as expected due to its lack of journaling features.

These results confirm that filesystem design significantly impacts forensic recovery potential.



Scenario 3: Metadata Tampering

FAT32 Results

Updated both mtime and ctime timestamps

Failed to detect tampering

No timestamp discrepancy to alert forensic investigators

NTFS Results

Updated mtime but preserved original ctime

Successfully detected tampering

Timestamp mismatch provides forensic evidence

ext4 Results

Updated mtime but preserved original ctime

Successfully detected tampering

Timestamp discrepancy reveals modification

We simulated a metadata attack by setting test2.txt's modification time to a future date (2026-01-01). Both NTFS and ext4 preserved the creation timestamp, creating a detectable discrepancy that would alert forensic investigators to potential tampering.

Forensic Insights and Recommendations

NTFS Strengths

Robust MFT/journal, excellent metadata forensics

ext4 Advantages

Strong journaling, Linux-native, reliable metadata

Future Work

Simulate structural corruption, use Autopsy for deeper analysis

FAT32 Limitations

Fast repairs but no journaling, weak tamper detection

Our proof of concept confirmed that NTFS and ext4 provide superior forensic capabilities compared to FAT32. For critical systems or those requiring forensic readiness, we recommend avoiding FAT32 entirely. One key limitation we discovered is that standard filesystem tools don't recover overwritten data - specialized forensic tools are needed for that level of recovery.

Thank you for your attention! We welcome any questions about our methodology or findings.