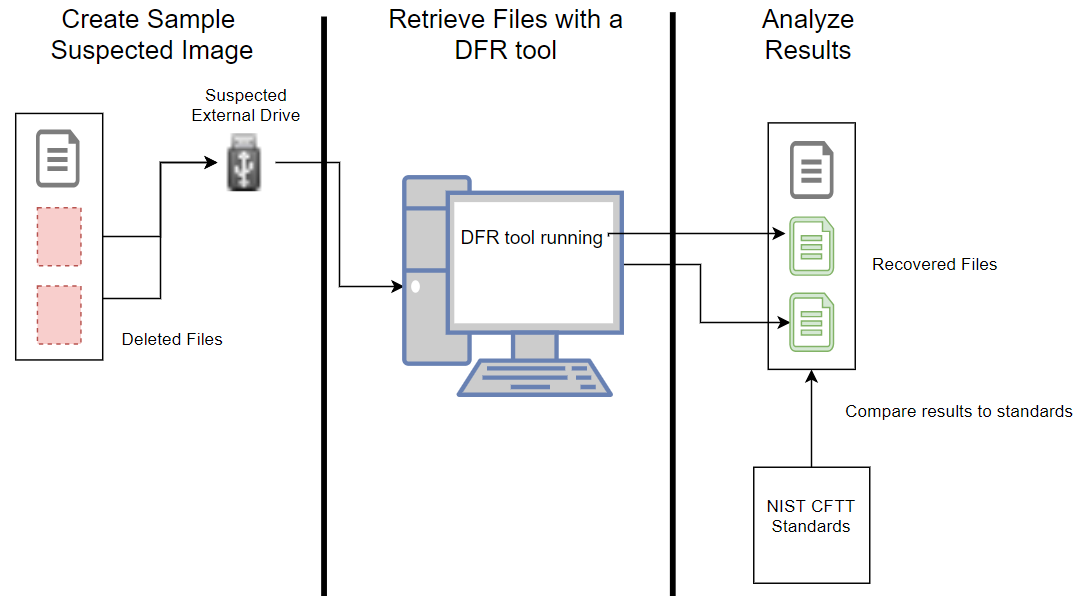
**Background**

Digital forensic tools are used for post-mortem investigation of cyber-crimes and cyber-attacks both in corporate and government organizations. National Institute of Standards and Technology (NIST)’s Computer Forensics Tool Testing Program (CFTT) established standards for digital forensic tools to help determine the quality and integrity of such tools. The quality and integrity of these tools are especially important in judicial proceedings. Using a forensic tool that does not follow the standards may cause evidence to be thrown out in court cases (which are increasingly reliant on digital data). Consequently, incorrect results from a forensic tool can also lead improper prosecution of an innocent defendant. The focus of our proposed research is about standardization of one class of forensic tools that are for Deleted File Recovery (DFR). The CFTT standard for DFR tools consists of four *core* features and a set of *optional* features

**Research question**

****A DFR tool is a software that can retrieve (residual data of) a file that was deleted from a storage device (e.g., computer hard disk, flash drive, and more). We plan to investigate a set of popular DFR tools to find which of these tools meet the standard set by CFTT. There are many companies and individuals marketing their software as the best recovery tool. As one task we will test the free tools’ effectiveness at recovering files compared to that of enterprise-level tools. Identifying if there are any errors for such as not recovering a deleted file or attempting to recover a file that was never there is also an important metric for a DFR tool (*Type I* and *Type II* errors). Type I and Type II errors of a tool can relate it to the standard. Using popular tool *dd* we can create a file system *image*. An image is a single file of the entire drive that can be used in a multitude of ways. In our case we will use the image to have consistent cases across the various DFR tools and upload to our Github (see appendix C for link) to use for future tests and research.

**Process**

We used the following process to create the cases into images that can be used against the DFR Tools to get results and compare them.

**Generating Test Cases:**

1. Zero out partition (make sure drive is entirely empty)
2. Use standard files (size and contents) to fill up flash drive with scenario
   1. Files were created in 1 Megabyte (MB) block sizes unless specified otherwise
3. Delete appropriate files that we are testing to be found
4. Create image of drive using dd tool

**Evaluating DFR Tools:**

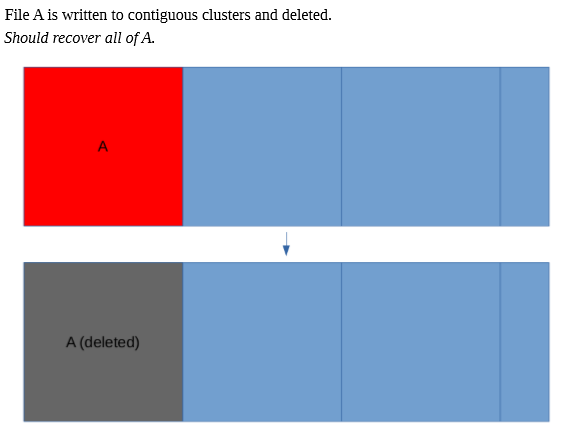
For each tool, the following process was followed

1. Mount image in Linux for DFR software to recognize and run against  
   **OR**
2. Use the image file directly in the DFR software
3. Export results and upload to GitHub for team
4. Analyze results compared to NIST Standards

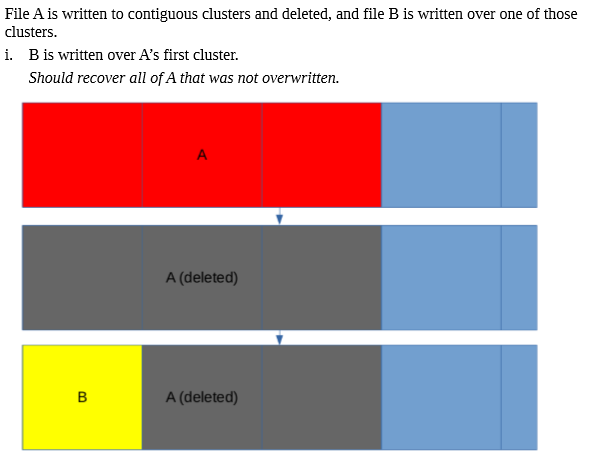
**Scenarios of test cases**

We created the cases with different scenarios in mind and possibilities of users deleting data. All diagrams can be found on Appendix B. Here are a few examples of very common instances that can show the importance of a reliable DFR tool.

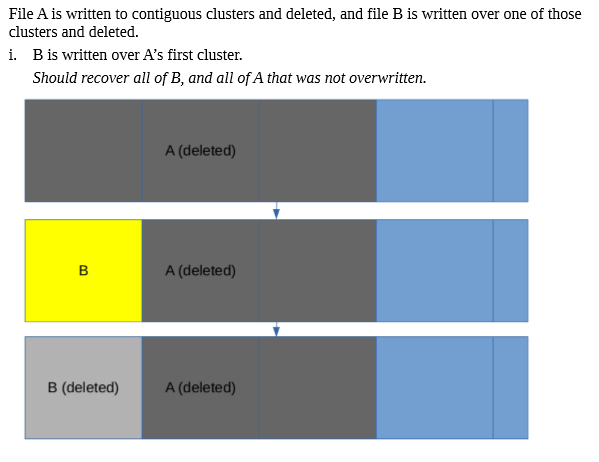
Case 1 (Data is deleted). This case is the base standard for all tools. The purpose of this case is to test all the tools to see if they identify and recover a single deleted file. Criminals or users can inadvertently delete critical data and believe it is gone forever.



Case 4i (Data is deleted and overwritten in hopes of not being found). This test is a bit more complicated, for both the tool and how it happens. It is possible to find methods and tools that promise creating a new file to ensure the deleted file is gone forever. By overwriting the first portion of the deleted file, the goal is to prevent the tool from recognizing A. However, according to NIST standards, the tools should be able to recover as much of file A as possible without recovering file B



Case 5i (Data is deleted and overwritten and deleted again in hopes of only the most recently deleted data is retrieved). Even more complicated than the previous example. By deleting file B that overwrote A, the purpose is to make the tools believe that file B is part of A. However, according to NIST standards, the tools should recover file A and file B separately.



**Results**

Results are shown for each tool per case. The results are broken up into a table to explain what was expected, what was recovered, and if the tool passed or failed the test.

Unless otherwise specified, files are named after creation (File A, File B, etc.) and are in 1Megabyte (MB) sizes. All diagrams of cases can be found in Appendix B.

Example Table: *expected results to be recovered*

|  |  |  |  |
| --- | --- | --- | --- |
| Case ## Results | | | |
| DFR Tools: | DFR tool #1 | DFR tool #2 | DFR tool #3 |
| File \_\_ Result: | Details about file recovered | Any data in file not expected | Incorrect recoveries are marked in red for easy viewing |
| File \_\_ Result: | Separate rows for each file expected to be recovered | Missing or mislabeled files are marked red for easy viewing | Incorrect recoveries do not affect other rows of same tool |

*Notes about results. Any possible explanations of unpredicted behavior.*

**FAT**

Case 1: *recover file A*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case 1 Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered A | Recovered A | Recovered A | Recovered A | Recovered A | Recovered A |

*All tools recovered the file correctly.*

Case 2:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case 2 Results | | | | | | |
| DFR Tool | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered all of A | Recovered all of A | Recovered first half of A | Recovered first half of A plus data from B | Recovered first half of A plus data from B | Recovered all of A |

*Even though a tool only recovered the first half of file A, we consider that passing. The reason we consider this passing is because file A was split and the tool should be able to determine at least where the first section ends. However, Magnet Axiom and Recuva recovered file B as part of File A which is why they are marked as failed in this test*

Case 3:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case 3 Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered first half of A plus data from B | Recovered first half of A plus data from B | Recovered first half of A plus data from B | Recovered first half of A plus data from B | Recovered first half of A plus data from B | Recovered first half of A plus data from B |
| File B result: | Recovered B | Recovered B (separate file) | Recovered B (separate file) | Recovered B (separate file) | Recovered B (separate file) | Recovered B (separate file) |

*Every tool was able to recover file B separately. However, every tool failed to recover the proper data of file A*

Case 4i:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case 4i Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered 2kb of file B | Recovered 2kb of file B | Empty file (0 Bytes) | Recovered size of A, but data was B and then zeros | Recovered size of A, but data was B and then zeros | Recovered B and last portion of A |

*Autopsy was unable to recover any relevant data. FTK Imager was unable to recover any data even though it saw file A was deleted. For Magnet Axiom and Recuva we believe it could not find any data, so it substituted zeros in the rest of the size of A*

Case 4ii:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case 4ii Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered 2kb of zeros | Recovered 2kb of zeros | Recovered 1MB of A | Recovered 1MB of A, 1MB of B, then 1MB of Zeros | Recovered 1MB of A, 1MB of B, then 1MB of Zeros | Recovered 1MB of A, then 2MB of zeros |

*Even though FTK Imager only recovered half of file A, we consider it passing because it did not try to retrieve file B like Magnet Axiom and Recuva did. Test disk also is considered passing because it understood the beginning and end of file A without putting file B in between*

Case 4iii:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case 4iii Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered 2kb of B | Recovered 2kb of B | Recovered 0kb file | Recovered B, 1 kb of zeros and 1kb of A | Recovered B, 1 kb of zeros and 1kb of A | Recovered B, 1 kb of zeros and 1kb of A |

*The reason Magnet Axiom, Recuva, and Testdisk all recovered 1kb of zeros was because of the size of B. Because B did not finish filling the whole section when it was created the filesystem overwrote file A and filled it with zeros.*

Case 4iv:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case 4iv Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered 2kb of B | Recovered 2kb of B | Recovered 0kb file of A | Recovered B | Recovered B | Recovered B |

*FTK Imager is considered passing in this case because file A was completely overwritten. FTK imager recognized that File A existed at one point, but did not try to substitute File B’s data in the recovered File A*

Cave 5i:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case 5i Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered B and rest of A | Recovered B and rest of A | Recovered B and rest of A | Recovered B and rest of A | Recovered B and rest of A | Recovered B and rest of A |
| File B result: | Recovered B | Recovered B | Recovered B | Recovered B | Recovered B | Recovered B |

Case 5ii:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case 5ii Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered B in the middle of A | Recovered B in the middle of A | Recovered B in the middle of A | Recovered B in the middle of A | Recovered B in the middle of A | Recovered B in the middle of A |
| File B result: | Recovered B | Recovered B | Recovered B | Recovered B | Recovered B | Recovered B |

Case 5iii:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case 5iii Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered B, 1kb of zeros, and 1kb of A | Recovered B, 1kb of zeros, and 1kb of A | Recovered B, 1kb of zeros, and 1kb of A | Recovered B, 1kb of zeros, and 1kb of A | Recovered B, 1kb of zeros, and 1kb of A | Recovered B, 1kb of zeros, and 1kb of A |
| File B result: | Recovered B | Recovered B | Recovered B | Recovered B | Recovered B | Recovered B |

Case 5iv:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case 5iv Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered B as A | Recovered B as A | Recovered B as A | Recovered B as A | Recovered B as A | Recovered B as A |
| File B result: | Recovered B | Recovered B | Recovered B | Recovered B | Recovered B | Recovered B |

Case 6:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case 6 Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered 2kb file of zeros | Recovered 2kb file of zeros | Recovered first half of A | Recovered first half of A, and B as 2nd half | Recovered first half of A, and B as 2nd half | Recovered first half of A, then 1MB of zeros |

Case 7:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case 7 Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered first half of A, and C | Recovered first half of A, and C | Recovered first half of A | Recovered first half of A | Recovered first half of A, and B | Recovered first half of A, and C |
| File C result: | Recovered C | Recovered C | Recovered C | Recovered C | Recovered C | Recovered C |

Case 8:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case 8 Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered 2kb of zeros in A’s name | Recovered 2kb of zeros in A’s name | Recovered first half of A | No file was recovered for A | Recovered half of A and 52kb of zeros | Recovered half of A and 4kb of zeros |

Case 9:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case 9 Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered 2kb of zeros in A’s name | Recovered 2kb of zeros in A’s name | Recovered first half of A | No file was recovered for A | Recovered half of A and 52kb of zeros | \*\*\*  Recovered C |

*The result of Testdisk we believe to be an error in the process of testing the tool for this individual case. Case 9 on Testdisk will be reviewed and retested to ensure the correct process*

Case 10:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case 10 Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered 2kb of zeros in A’s name | Recovered 2kb of zeros in A’s name | Recovered first half of A | No file was recovered for A | Recovered half of A and 52kb of zeros | Recovered half of A and 4kb of zeros |
| File B result: | Recovered B | Recovered B | Recovered B | Recovered B | Recovered B | Recovered B |

**NTFS**

*Note: some cases have not been run against magnet Axiom yet, there were delays in converting the images that did not allow time to run the tests and report on them.*

Case 1:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered A | Recovered A | Recovered A | Recovered A | Recovered A | Recovered A |

*All tools recovered the file correctly.*

Case 2:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered both parts of A | Recovered both parts of A | Recovered both parts of A | Recovered both parts of A | Recovered both parts of A | Recovered both parts of A |

*All tools were able to recover both parts of file A even though it was broken up*

Case 3:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered both parts of A | Recovered both parts of A | Recovered both parts of A | Recovered both parts of A | Recovered both parts of A | Recovered both parts of A |
| File B result: | Recovered B | Recovered B | Recovered B | Recovered B | Recovered B | Recovered B |

*Every tool was able to recover both files. Compared to FAT, none of the tools were able to recover file A*

Case 4i:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered B as part of A | Recovered B as part of A | Recovered zeros in the overwritten section, then the rest of file A | Test was not run yet | Recovered B as part of A | Recovered B as part of A |

*We were not able to convert all the images for Magnet axiom yet. We are still working on converting and running the separate images for Magnet Axiom and will update this document as we get results.*

Case 4iii:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered B, 2kb of zeros, then the rest of A | Recovered B, 2kb of zeros, then the rest of A | Recovered zeros, and the non-overwritten part of A | Test was not run yet | Recovered B, 2kb of zeros, then the rest of A | Did not recover any files |

Case 4iv:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered B as A | Recovered B as A | Recovered zeros | Test was not run yet | Recovered B as A | Did not recover any files |

Case 5i:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered B as part of A | Recovered B as part of A | Recovered B as part of A | Test was not run yet | Recovered B as part of A | Recovered B as part of A |
| File B result: | Recovered B | Recovered B | Recovered B |  | Recovered B | Recovered B |

Case 5iii:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered B, 2kb of zeros, then the rest of A | Recovered B, 2kb of zeros, then the rest of A | Recovered B, 2kb of zeros, then the rest of A | Test was not yet run | Recovered B, 2kb of zeros, then the rest of A | Recovered B, 2kb of zeros, then the rest of A |
| File B result: | Recovered B | Recovered B | Recovered B |  | Recovered B | Recovered B |

Case 5iv:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered file B as file A | Recovered file B as file A | Recovered file B as file A | Test was not yet run | Recovered file B as file A | Recovered file B as file A |
| File B result: | Recovered file B | Recovered file B | Recovered file B |  | Recovered file B | Recovered file B |

Case 6:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered first half of A and file C as part of A | Recovered first half of A and file C as part of A | Recovered first part of A, then zeros | Test was not yet run | Recovered first half of A and file C as part of A | Recovered first half of A and file C as part of A |

Case 7:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Case Results | | | | | | |
| DFR Tool: | Autopsy (Linux) | Autopsy (Windows) | FTK Imager | Magnet Axiom | Recuva | TestDisk |
| File A result: | Recovered first half of A and file C as part of A | Recovered first half of A and file C as part of A | Recovered first half of A and file C as part of A | Test was not yet run | Recovered first half of A and file C as part of A | Recovered first half of A and file C as part of A |
| File C result: | Recovered file C | Recovered file C | Recovered file C |  | Recovered file C | Recovered file C |

**Larger picture**

Based on the data collected, a few conclusions can be made. DFR tools had better results when run on NTFS file systems instead of FAT. This can be because of how the different file systems store the metadata surrounding a file. In NTFS it is easier for DFR tools to determine what is a deleted file, and what should not be recovered. However, even in NTFS a stronger conclusion can be made that none of the DFR tools tested meet the core standards as defined by NIST. Only in the easiest and simplest cases do all tools properly recover the file. For a more meaningful conclusion, there are two possibilities. Either the tools are not good enough, or the standards themselves are insufficient. The majority of failures were on the grounds of the third and fourth core features. DFR-CR-03 states that a tool must recover “all non-allocated data blocks identified in a residual metadata entry.” In other words, all sections of the drive indicated by file system structures as belonging to the deleted file, besides those which have been reassigned to a new file, should be recovered. This is reasonable; however, the meaning varies depending on what is considered “residual metadata,” which differs based on the file system. For example, in the FAT file system, the only information about a deleted file is the file size and starting location on the disk. If the file is fragmented into multiple parts, there is no direct evidence of it. In this situation we determined that only the first fragment must be recovered, because residual metadata gives no information about fragmentation. DFR-CR-04 states that a tool must only recover “data blocks from the Deleted Block Pool.” In other words, all of the data in the recovered file must be from the original deleted file. There are some situations in which this is impossible for the tool to determine with certainty. The test cases 5i-5iv illustrate this. When a deleted file is overwritten by a second file, and then the second file is deleted as well, there is no way to tell which file overwrote which. Both FAT and NTFS store time information about files; however, that information is non-essential metadata, meaning it is not necessary for the operation of the file system and can thus be altered without consequence. With just the essential metadata, in this case simply the locations of each file on the disk, the tool must guess which file came first to avoid violating the fourth core feature. If it makes a conservative estimate and only recovers the parts of each file that don’t overlap the other, it would fail to fulfill the third core feature. File recovery is not an exact science, and there are many cases where metadata is simply not enough to properly recover a file. For the standards to be realistic, they must be more specific about what constitutes “residual metadata” for various types of file systems and resolve situations where the third and fourth core features will be in conflict.

**Appendix**

1. **How test cases were created**

Test procedure:

In these steps I use only command line tools for any step that modifies the file system, so the methods used to create the image can be easily documented and reproduced. All those steps are performed in the VM for the same reason.

Test files are each a sequence of the same ASCII character, of a set length. For example: aa1M is 1 MiB of ‘a’.

To prevent Ubuntu from automatically mounting filesystems: https://help.ubuntu.com/community/Mount/USB#Configuring\_Automounting

1. Partition USB drive (host or guest OS) I used gdisk for this, and made several 4MiB partitions

For each test image: (example commands used for creating test case 1) 2. Zero over partition (host or guest OS) dd if=/dev/zero of=/dev/sdx1

3. Write file system (guest OS) mkfs.fat -n CASE\_1 /dev/sdx1

4. Mount file system (guest OS) mnt /dev/sdx1 /mnt

5. Perform and document file writes and deletions for given test case (guest OS) Unmount and remount file system before deletions to ensure buffered writes actually make it to the disk cp aa1M /mnt umount /mnt mnt /dev/sdx1 /mnt rm /mnt/aa1M

6. Unmount file system (guest OS) umount /mnt

7. Make image of file system (host OS) dd if=/dev/sdx1 of=/.../case\_1.raw

8. Manually inspect file system image to verify it fits the test case (host or guest OS) Using a hex editor or similar tool for viewing raw data

1. **Test Case Diagrams**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a social media post

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

1. **Github**

The team used Github as an online repository for multiple uses. Being able to sync our data and results to each other and tracking of reports and other documents. All images and steps referenced in this report can be found at the following Github link

<https://github.com/quintoc/CURS-DFR-Research>