Assignment 2 - Tool Testing

Kennet Fladby, Christopher Fullu, Lars Olav Gigstad, Rune Hammersland
{firstname.lastname} @ hig.no

November 30, 2007

Abstract

Project description: Research a tool with respect to potential use in digital forensics and forensic soundness. Perform practical experiments and write a report based on the results. You may use existing evidence images (e.g., scan of the month, forensic challenge, or honey pot images) or you may create a new evidence image.

1 Introduction

Today, computers are used to an extent where everything is virtually connected to everything. This implies that a lot of criminal activity also occurs on computers in many different forms. The need for forensic examination of computers is increasing and can often provide important clues to solve a case [1]. When such an examination is done it is important that the evidence is kept in its original state (just like physical evidence) in such a way that the content is not altered in any way. If a hard drive is acquired as evidence, the examination is done on images of the disk rather than the disk itself. One important consideration is that the images have be an exact copy of the original disk down to every last bit, in addition to no alterations or adding of data, called *forensic soundness*. This also applies to the tools that are used in the examination process. For this project we will compare tools to recover deleted and corrupted files. It would be interesting to simulate even more errors by using a system as explained in [2], but considering the time line of the project we limit ourself to a few selected scenarios. The project will be focused on how well the tools perform and whether they meet the requirements of forensic soundness or not. Very much like what they do in [3]. The tools we will be looking at are:

- Fatback [4] - A *nix tool for recovering files from FAT file systems.

- ntfsundelete [5] - recover a deleted file from an NTFS volume. ntfsundelete only ever reads from the NTFS Volume. ntfsundelete will never change the volume.
- R-Studio [6] - This is a data recovery software for Windows that supports a whole range of file systems: FAT (all variants), NTFS, HFS/HFS+, EXT2/3 and UFS1/2 to name a few.
- GetDataBack [7] - GetDataBack is an application to recover data on hard drives where the partition table, boot record, FAT/MFT or root directory are lost or damaged, even if the file system is no longer recognized by windows. It is a read-only application that will never write to the disk in process of being recovered.
- Magic-Rescue [8] - Magic Rescue scans a block device for file types it knows how to recover and calls an external program to extract them. It looks at "magic bytes" in file contents, so it can be used both as an undelete utility and for recovering a corrupted drive or partition.

2 Project set-up

Windows XP and Debian GNU/Linux were installed in VMWare. An extra FAT- and NTFS partition of 50 MB were created for the Windows machine and one extra EXT2 partition of 50 MB for the Linux machine. For simulation purposes we prepared some files to be copied onto the partition:

1. `hello.jpg`: a JPEG image of a kitten (see Appendix A.1).
2. `hello.mp3`: an MP3 file.
3. `hello.txt`: a plain text file containing the strings `ALL NIGHT` and `POST MOR PICZ OF TEH KITTANZ` (and more, see Appendix A.2).

The files were hashed using MD5 to determine whether the they get damaged during recovery or not to ensure that integrity is achieved:

hello.jpg 36e7f851d472cf190446603f26a739e6

hello.txt 40a81b8292c5f51c7cd48fc4c4556aa0

hello.mp3 713700fa8a35aa53be7bce1fc11003b9

The files were stored on all the extra FAT-, NTFS- and EXT2 partitions, along with some other JPEG images. Using `dd`, we created an image of the partition in the following way (example is for Linux, similar options were used for Windows): `dd if=/dev/hdXn of=ext3-1.dd conv=noerror,sync`. The last option guarantees that `dd` will continue if it encounters an error, and pad the block containing the error with NULs until it reaches the block size. This way an error in a block in the file system will only affect that block, and not propagate throughout the image.

The images were named after the file system in question, and -1 was appended in order to enumerate later images of the same file system. Then we created an MD5 hash from the partitions (example again from Linux): `md5sum /dev/hdXn > ext3-1.md5`, and verified that the hash from the image matched the hash from the partition. The hashes were named the same way as the images, only with a `.md5` suffix in stead of `.dd`.

Having completed this step, we deleted the files from the partition, and created new images and new MD5 sums, following the aforementioned procedure. This time the number in the filenames were incremented to 2. The partitions were then formatted using quick format under Windows and `mkfs.ext2` under Linux, and we created a new set of images and MD5 sums (with the index 3). Finally we we did a full format of the FAT and NTFS partitions, and created a new set of images and MD5 sums. The snapshot feature in VMWare was used to restore the state of the Linux host to the first step (after having copied the files), and then we first "zeroed" the partition, using `dd if=/dev/zero of=/dev/hdXn`. Then we reverted to the same step, and tried to secure erase the partition, using `dd if=/dev/urandom of=/dev/hdXn`. Initially we tried to use `/dev/random` as this device produces better random numbers (see `man 4 random`), but this took too much time, so we used `/dev/urandom` in order to complete the project in a timely fashion. Again we created a disk image and a corresponding MD5 sum.

To recap:

1. Copied the files to the partition.
2. Deleted the files.
3. Used quick-format on the partitions.
4. Did a full format of the Windows partitions.
5. Reverted to step 1, and "zeroed out" the EXT2 partition.
6. Reverted to step 1, and "shredded" the EXT2 partition.

Having completed these steps, we had 12 files (4 per file system) of the format "file system-iteration.dd" and 12 files with the same naming scheme but with

`.md5` as the file suffix. All files were uploaded to a server for practical storage reasons.

3 Investigation

When working with the files, we booted a Helix image and downloaded the files to a temporary folder. The FAT images and MD5 sums were put in a folder called `fat`, and similar was done for NTFS and EXT2.

3.1 FAT

Using fatback, we issued the following command to retrieve files from the image: `fatback -o recovered/ -a -s fat-2.dd`. For some reason the -o option (which specifies a folder to place recovered files in) did not work, and all files landed in the current directory. The -s option specifies that we have a single partition, which is true for us, since we didn't image the whole disk, only the relevant partition. After recovering the files, the MD5 sum of the image were checked again to make sure the tool didn't tamper with the image itself.

We repeated this process with the rest of the images, but sadly fatback relied upon the file allocation table for finding deleted files, so after a format of the partition (where the allocation table is wiped and replaced with a blank one) fatback were no longer able to recover the deleted files, even though the files still were on the drive for some of the images.

Using R-Studio, we imported the disk images one after one, and proceeded to scan them. After scanning a disk image, R-Studio will indicate possible starts of file systems within that image. Right clicking on the first detected file system (which was the correct guess for all images we used), you get an option to show files on the system. With R-Studio, we managed to recover the deleted files in all the FAT images.

GetDataBack delivered weak results: it didn't even find all files on the image where they were simply deleted. On the formatted images it had even greater problems in locating the files we know to exist on the partition. Most of the files were nowhere to be found: only a couple of image files were recovered.

3.2 NTFS

With NTFSundelete the command `ntfsundelete ntfs-2.dd -s` will scan the file system and provide a list over files found on the system. To recover the files the command `ntfsundelete ntfs-2.dd -u inode -o filename` is used, where `inode` is an inode found during the scan, and `filename` is the name of the deleted file. The term "inode" is usually not used about NTFS, but the man page for ntfsundelete used this term (which probably means the developers are more familiar with the EXT file system).

ntfsundelete managed to recover the files in the image where they were simply deleted. When the partition had been formatted, ntfsundelete could no longer find the deleted files. This is because it looks for deleted "inodes" and not blocks of data that resembles known file types. However R-Studio managed to find the files on all NTFS images, except the fully formatted. In the last case it found many of the files, but the JPEG image of the kitten was lost.

Again, GetDataBack disappointed us, as it only managed to find some of the files. One might argue that it does a better job of finding files than ntfsundelete and fatback, as it still finds files when the allocation table has been wiped, but when it doesn't find the files when "inodes" still are lying around, we are not pleased with the results.

3.3 EXT2

The EXT2 images presented more problems than their FAT and NTFS counterparts. The intention was to use e2recover to recover the files, but we had several problems in finding out exactly how to use this tool. The man page for the program states that it takes input from a log file made by issuing `ls -d` in the debugfs-program. However this approach didn't seem to do anything for us. We tried to feed the input to e2recover through a file, and by feeding it directly from STDIN. After some trial and error, we gave up on e2recover and used a program called Magic Rescue.

Magic Rescue scans the image, or block device, for blocks of data which resembles predefined "recipes". The program comes with some predefined recipes for finding JPEG-JFIF files, JPEF-EXIF, MP3 files, and more can be added. By issuing the command `magicrescue -r jpeg-jfif -r mp3-id3v1 -d recovered/ ext2-2.dd` where recipes are listed as -r options, and the -d option indicates where to store the recovered files, we were able to recover the MP3 file and the JPEG image. Sadly the image was somewhat corrupt. It had correct size, and roughly the top half of it seemed correct, but the bottom half were skewed, had distorted colors, and some other artifacts. Magic Rescue also worked after for-

matting the partition, but not after zeroing it and certainly not after overwriting it with random data.

Examining the EXT2 images in R-Studio yielded the same result for the image of the kitten, and there were problems in recovering the text file. The program found several small text files containing something resembling a character, but most of it didn't seem to fit with the original file, so it's hard to say if we would have been able to piece it together. This might be because of fragmentation of the file, but it is very unlikely, as the three files we were looking for were the first files to be put on the partition (and the file system should be able to find a continuous block of unallocated space).

GetDataBack doesn't support EXT2. None of the tools found any files on the image where the partition had been zeroed out, and none of the tools found any files when the partition had been overwritten with random data.

4 Results

Even though we found the files on some of the images, and with some of the tools, the MD5 sum didn't always match. It obviously didn't match for the corrupted image we recovered, but we had some additional problems. The text files recovered under Windows using R-Studio and GetDataBack didn't match the MD5 sum, even though the contents was the same. The reason for this is that for convenience, the hello files were stored on a web server, and downloaded to the hosts. When we did this on Windows, the text file was stored with CR-LF line endings, while the Linux host stored it with LF line endings. The MD5 sum reported at the beginning of this report is created from a version with LF line endings. We noticed this slip late in the project phase.

The image of the kitten (recovered from the FAT and NTFS images) also had differing MD5 sums from the original. To the naked eye, the recovered image looks identical to the original image, and they do have the same data to a certain extent. However one of them contains some null data at the end which the other doesn't have. In other words: The recovered files we have does not hold up against the principle of forensic soundness. See Table 1 for a summary of how the different programs performed.

Table 1: Legend: $\surd$: OK. $\otimes$: Not OK. $\star$: some files. N/A: Not Applicable.

	Fatback	ntfsundelete	GetDataBack	R-Studio	Magic Rescue
Deleted Files	$\surd$	$\surd$	$\star$	$\surd$	$\star$
Quick Format	$\otimes$	$\otimes$	$\star$	$\surd$	N/A
Format	$\otimes$	$\otimes$	$\star$	$\star$	$\star$
Zeroed	N/A	N/A	N/A	$\otimes$	$\otimes$
Randomized	N/A	N/A	N/A	$\otimes$	$\otimes$

5 Conclusion

This project might have benefited from a slightly different angle: Trying to "securely" erase a file using several different programs / approaches, and then trying to recover the file using a program known to find files on a given file system regardless of corruption. Then we could have focused on one file system, and one tool for the hard task of recovering a file, while using several tools for the easier part. File recovery tools often have a wide range of options to be figured out, while different programs for erasing of files might only take the file- or device name as a parameter.

It seems like the EXT2 file system was the hardest system to recover files from. Both R-Studio and Magic Rescue had problems recovering the image of the kitten: both produced similarly corrputed images. As no additional writing to the partition was done between deleting the image and creating a new image of the partition, the problem shouldn't stem from additional writes to the partition. We are not sure what is causing the problem, but it seems to be related to the file system – not the tools used – as both tools produced very similar results. When it comes to the text file, Magic Rescue had no recipe for restoring text files, so only R-Studio managed to recover fragments of text files. However we are not sure if piecing them together would make sense. Using `strings` on the images, we managed to find the text file in both `ext2-2.dd` and `ext2-3.dd`, so it seems strange that R-Studio was not able to find it.

The results from FAT and NTFS were better, but when recovering files from a formatted volume, the md5sums did not match for all the files anymore. This means that the files in question no longer can be considered forensically sound. The difference between the original files and the recovered ones are not that large, besides some zero's added at the end of the file, but still this is not to be considered the same file anymore. A full format gave us problems on FAT/NTFS as well, fragments of the text file, the kitten JPEG not found, just to state a few problems. This indicates that other measures need to be taken to recover files from a fully formatted volume.

References

[1] James A. Whittaker and Michael Howard. Computer forensics. *IEEE SECURITY & PRIVACY*, July/August 2005.

[2] S. Avramov-Zamurovic, J. Lyle, and C. Wick. Hard disk interface used in computer forensic science. *Instrumentation and Measurement Technology Conference, 2004. IMTC 04. Proceedings of the 21st IEEE*, 3.

[3] Christopher V. Marsico and Marcus K. Rogers. ipod forensics. *International Journal of Digital Evidence*, 4, Fall 2005.

[4] Fatback. `http://sourceforge.net/projects/fatback/`.

[5] ntfsundelete. `http://www.linux-ntfs.org/doku.php?id=ntfsprogs`.

[6] R-studio. `http://www.data-recovery-software.net/`.

[7] Getdataback. `http://www.runtime.org/gdb.htm/`.

[8] Magic-rescue. `http://jbj.rapanden.dk/magicrescue/`.

Appendix

A Files

A.1 hello.jpg

A.2 hello.txt

```
Date: 10/28/2001 11:45 PM
From: Ian Arnold
Subject: !!!

OMGOMGOMG TEH CAT IS SI TEH EROTIK I WENT FAP FAP FAP FAP FAP FAP ALL NIGHT
LONG TO IT!!!1
PLZ POST MOR PICZ OF TEH KITTANZ BTU NOEN OF TEH DOGGIE TEH DOGGIE
SUXORZZORZZORZZZ!!!111
GODBYE SIRZ!!!!!1
```