Gernot Krautberger, Fotolia

Carving tools help you recover deleted files

UNDELETED

Modern filesystems make forensic file recovery much more difficult. Tools like Foremost and Scalpel identify data structures and carve files from a hard disk image. **BY RALF SPENNEBERG**

IT experts and investigators have many reasons for reconstructing deleted files. Whether an intruder has deleted a log to conceal an attack or a user has destroyed a digital photo collection with an accidental *rm -rf*, you might someday face the need to recover deleted data. In the past, recovery experts could easily retrieve a lost file because an earlier generation of filesystems simply deleted the directory entry. The meta information that described the physical location of the data on the disk was preserved, and tools like The Coroner's Toolkit (TCT [1]) and The Sleuth Kit (TSK [2]) could uncover the information necessary for restoring the file.

Today, many filesystems delete the full set of meta information, leaving the data blocks. Putting these pieces together correctly is called file carving – forensic experts carve the raw data off the disk and reconstruct the files from it. The more fragmented the filesystem, the harder this task become.

Many open source tools automate the carving process: The list is headed by Foremost [3] and its derivative Scalpel [4], but other tools include PhotoRec [5] and FTimes [6]. PhotoRec does not support generic carving for any file type, and FTimes is so hard to use it is not worthwhile for most users.

Foremost and Scalpel are not interested in the underlying filesystem. They simply expect the data blocks of the files to reside sequentially in the image under investigation. The tools will find images in *dd* dumps, RAM dumps, or swap files. Carving will help to identify and reconstruct files on corrupt filesystems, in slack space, or even after installation of a new operating system, as long as the required data blocks still exist.

Of course, none of these tools can perform miracles, and they are not designed to retrieve data from physically damaged hard disks. Also, the carving process cannot access data blocks that have been overwritten.

Because carving tools do not rely on the filesystem, they need other sources of information to discover where a file starts and ends. Fortunately, many file types have known structures. The header and footer are often all that is needed to identify the file type and location. The Linux *file* command also uses header and footer information to identify file types.

File carvers investigate the whole hard disk, or disk image, to locate known headers and footers. They then carve out the blocks between the header and footer and store the data as a new file.

Listing 1: Configuration

```
gif  y  155000000  \x47\x49\x46\x38\x37\x61  \x00\x3b
gif  y  155000000  \x47\x49\x46\x38\x39\x61  \x00\x00\x3b
jpg  y  20000000   \xff\xd8\xff\xe0\x00\x10  \xff\xd9
jpg  y  20000000   \xff\xd8\xff\xe1\xff\xd9
jpg  y  20000000   \xff\xd8                  \xff\xd9
```

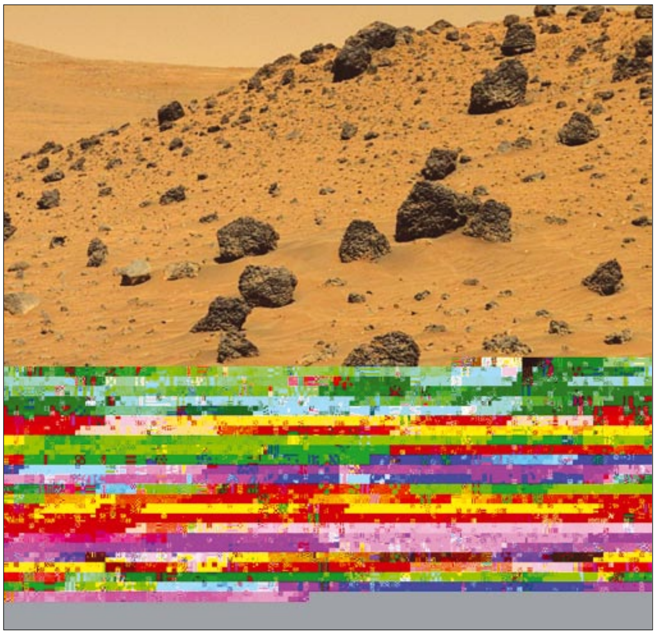

Figure 1: File carvers ignore the filesystem and carve the images directly from data blocks. In cases of fragmented files, the carver returns an imperfect photo, but this image might be sufficient to identify the subject.

Some file types do not possess unique footers. Carvers will at least guess where the file ends on the knowledge of where the next header starts. Of course, any amount of unidentified data could reside between the end of the file and the next header.

To avoid collecting unnecessary junk data, carving programs allow users to set maximum file sizes. Unfortunately, headers and footers are often short, which leads to numerous false positives.

Image formats are an exception. For example, each JPEG file starts with a byte sequence of *0xFFD8*, typically followed by *0xFFE00010*. File carvers are thus very good at identifying JPEG images. However, if some blocks have been overwritten, or if the file is fragmented, the tools will restore only a part of the file at best (Figure 1).

Foremost and Scalpel

Jesse Kornblum and Kris Kendall from the United States Air Force Office of Special Investigations developed Foremost in March 2001 as a tool for analyzing and recovering deleted files. The Foremost carving tool is inspired by an earlier program called CarvThis, which was created back in 1999 by Defense Computer Forensic Lab but never released to the general public. Foremost is now open source, and Nick Mikus maintains the source code after giving the program a major boost in the scope of his Master's degree.

Golden G. Richard III developed a separate program dubbed Scalpel based on Foremost 0.69. For a long time, Scalpel was regarded as an advanced tool. Some sources even claim that the Foremost developers recommend Scalpel themselves [7]. To be more accurate, both projects are under active development. Although Scalpel was far superior to its predecessor in 2005 – with the ability to analyze images around 10 times faster – Foremost has caught up recently

Listing 2: Foremost Run

```
Foremost version 1.5.3 by Jesse Kornblum, Kris Kendall, and Nick Mikus
Audit File

Foremost started at Sat Feb  9 18:36:29 2008
Invocation: ./foremost -v -T -i ../dfrws-2006-challenge.raw
Output directory: /linux-magazin/foremost/foremost-1.5.3/output_Sat_Feb__9_18_36_29_2008
Configuration file: /linux-magazin/foremost/foremost-1.5.3/foremost.conf
Processing: ../dfrws-2006-challenge.raw
|----------------------------------------------
File: ../dfrws-2006-challenge.raw
Start: Sat Feb  9 18:36:29 2008
Length: 47 MB (49999872 bytes)

Num    Name (bs=512)       Size    File Offset    Comment

0:     00003868.jpg       280 KB       1980416
1:     00008285.jpg       594 KB       4241920
2:     00011619.jpg       199 KB       5948928
3:     00012222.jpg         6 MB       6257664
[...]
20:    00045015.zip       274 KB      23047680
21:    00007982.png         6 KB       4086865    (1408 x 1800)
22:    00033012.png        69 KB      16902215    (1052 x 360)
23:    00035391.png        19 KB      18120696    (879 x 499)
24:    00035431.png        72 KB      18140936    (1140 x 540)
*|
Finish: Sat Feb  9 18:36:32 2008

25 FILES EXTRACTED

jpg:= 11
htm:= 5
ole:= 2
zip:= 3
png:= 4
------------------------------------------------------------------

Foremost finished at Sat Feb  9 18:36:32 2008
```

thanks to Nick Mikus, and it is actually superior to its derivative for some tasks.

Both Foremost and Scalpel use configuration files to specify which files to search for (Listing 1). The first column designates the file type and also specifies the file extension to add to any files the program finds. Files for which the case is relevant in the header and footer have a *y* in column two; this is *n* for all others. The next column defines the maximum file size, followed by the header byte sequence, and the footer byte sequence if it exists. The *\x* string introduces a byte in hexadecimal notation; the other possibilities are *\s* for a space and *?* as a wildcard for any character. Other options can follow at the end.

Fast Finder

Because of its origins, Scalpel uses the same configuration file as Foremost, although the two tools work differently internally. Both tools find more or less the same files, but there are some discrepancies in file identification. Forensic experts are thus well advised to use both programs.

Versions 0.9.1 and later of Foremost use a new approach to identifying ZIP, JPEG, Office, and other formats. The formats are implemented directly in Foremost, meaning that the program does not need header and footer information in the configuration file for the identification process. Foremost enables this new detection function if you set the *-t* flag at the command line followed by the required file types:

```
foremost -T -t jpg,gif,pdf ⮐
-i imagefile
```

Supported formats are listed in Table 1. To enable all of these built-ins, just set the *-t all* option. The previous command line also sets the *-T* option to tell Foremost to write any files it finds to a directory that uses a name with a timestamp. This makes it easier to organize the forensic investigation, in that each new run writes its results to a new directory.

Space Requirements

The possibility of false positives means that the carver identifies a huge amount of data, so make sure you have enough free space on the target filesystem. The carving process doesn't necessarily require large amounts of copying. Virtual filesystems, such as CarvFS [8], are designed to access the data directly from the original image. CarvFS, which is based on FUSE (Filesystem in Userspace), only expects the carving tool to provide a table that describes which files are available at which physical locations. The CarvFS filesystem originated with the Dutch police's Open Computer Fo-

INFO

[1] The Coroner's Toolkit: *http://www.porcupine.org/forensics/tct.html*

[2] The Sleuth Kit: *http://www.sleuthkit.org*

[3] Foremost: *http://foremost.sf.net*

[4] Scalpel: *http://www.digitalforensicssolutions.com/Scalpel/*

[5] PhotoRec: *http://www.cgsecurity.org/wiki/PhotoRec*

[6] FTimes: *http://ftimes.sourceforge.net/FTimes/*

[7] Foremost on the Forensics Wiki: *http://www.forensicswiki.org/wiki/Foremost*

[8] OCFA, The carve path zero-storage library and filesystem: *http://ocfa.sourceforge.net/libcarvpath/*

[9] DFRWS carving challenge: *http://www.dfrws.org/2006/challenge/*

rensics Architecture (OCFA) project (see the article on OCFA in this issue), and it is intended for situations in which copying all the files to a separate location would result in huge volumes of data. In other cases, however, copying the data is more efficient than accessing it from the original image.

A typical Foremost run without built-ins is shown in Listing 2. The image for this example comes courtesy of the Digital Forensic Research Workshop (DFRWS [9]) challenge. DFRWS ran this competition in 2006 to test file carvers and promote their development. At the end of the competition, the organizers published a list of the files in the image.

PhotoRec

If the filesystem is not completely destroyed, tools that evaluate the filesystem provide an important alternative to tools such as Foremost and Scalpel. The PhotoRec [5] recovery tool was developed by Christophe Grenier to rescue photos from corrupt Flash memory. PhotoRec will also work if the partition table is damaged.

Once PhotoRec has identified the filesystem, it extracts an enormous variety of file types. In addition to photo files, PhotoRec also restores EXE or ZIP files.

All told, the tool supports more than 180 file types. The program is controlled by means of a practical text menu, which reduces the danger of user errors. Unfortunately, PhotoRec cannot current analyze RAM dumps or swap files.

Memory Hook

File carvers help forensic investigators extract deleted files. Foremost and Scalpel ignore the filesystem and can even restore data from RAM dumps and swap files. Their speed is quite amazing.

If the filesystem still exists, a tool such as PhotoRec is also useful for finding lost files. ■

Table 1: Foremost Built-ins

Format	Comment
Images	
JPG	JFIF, Exif, and RAW formats
GIF	Graphic Interchange Format
PNG	Portable Network Graphics
BMP	Windows bitmap files
Executables	
EXE	Windows PE, DLL, and EXE
Video and Audio	
AVI	Audio--Video Interleaved
MPG	Detects all MPEG files that start with 0x000001BA
WMV	Windows Media Video; WMA (Windows Media Audio) in part
MOV	Quicktime movie
Documents	
PDF	Portable Document Format
OLE	Object Linking and Embedding; for example, PowerPoint, Word, Excel, Access, Starwriter
DOC	Word files only
HTM	Hypertext Markup language (websites)
Archive formats	
ZIP	ZIP, JAR, MS Office 2007, Open Office 2.0 (zipped XML documents)
RAR	Roshal Archive
CPP	C source code; many false positives