The Linux Kernel and the Forensic Acquisition of Hard Disks with an Odd Number of Sectors

Jesse D. Kornblum
research@jessekornblum.com

Abstract

No official version of the Linux kernel, up through and including version 2.4, allowed a user land process to access the last sector of a hard disk or hard disk partition with an odd number of sectors. Although the inability to access this last sector did not affect normal operation of the system, it did prevent the complete forensic acquisition of such a disk. The author repeats an earlier experiment to verify the issue in version 2.4 of the kernel and then shows that the issue has been resolved in version 2.6. Systems using version 2.6 of the Linux kernel can completely forensically acquire disks or partitions with an odd number of sectors.

Introduction

The forensic acquisition of media refers to the process of making a bit-for-bit copy, or image file, of a piece of media. The image files created by forensic acquisition are frequently used in civil or criminal court proceedings, so great accuracy is required. Although many different tools are used by forensic examiners in the acquisition process, this paper uses only two of them, GNU dd and GNU md5sum, both part of the Coreutils project [3].

An evaluation of the GNU dd program conducted by the National Institute of Standards and Technology (NIST) [1] in 2002 noted that when dd was running on Linux, it failed to read the last sector of each disk or partition that had an odd number of sectors. That is, for a hard disk with *n* sectors, where *n* is odd, operating systems such as FreeBSD version 4.4 read *n* sectors, but RedHat Linux 7.1, running kernel version 2.4.2-2, read *n-1* sectors. Although some people thought the NIST report indicated a flaw in GNU dd, Dr. James Lyle of NIST identified the discrepancy as a function of the Linux kernel [2]. He wrote, "The problem is not in dd. Looking at the dd source code is a red herring; the source of the problem is in a software layer under dd, i.e., kernel or driver."

Hard disks normally store information in 512-byte sectors. Up through version 2.4, the Linux kernel would access data from the disk in blocks of 1024 bytes, or two sectors at a time. In general, the difference between block and sector sizes did not present a problem. If the disk had an odd number of sectors, there was no way for a user land process to access the last sector on the disk. (A user land process is an application such as a browser or word processor. A kernel land process is part of the operating system.) The kernel could access the last sector, but did not do so by default. Some

kernel patches to work around this problem were suggested [4], but these were unstable and not widely used in the forensics community.

Although the inability to access the last sector was not a problem for normal users, it was a concern for forensic examiners. Most Linux users never noticed that the last 512 bytes of disk space could not be accessed. Although the usable size of a disk with an odd number of sectors was reduced by 512 bytes, this was a small loss and could be easily overlooked. Only users who attempted to access the last sector on a drive with an odd number of sectors that had been formatted on another operating system would find themselves unable to read part of their data. But the flaw was an issue for forensic examiners, as they were using Linux systems to acquire drives formatted by other operating systems. Operating systems such as Microsoft Windows can use the last sector of a hard disk during normal operations; thus to get a complete image of the drive, an examiner must retrieve all of the data on the drive, including the last 512 bytes.

On 6 April 2002, Andries Brouwer contributed a patch [5] to the Linux kernel for reading large disks that, "Also solves other things, like the fact that Linux is unable to read the last sector of a disk or partition with an odd number of sectors." The patch was incorporated in the 2.5 series of the Linux kernel, or the development track, which later became the 2.6 version of the kernel. The author asserts that the patch submitted by Brouwer allows version 2.6 of the Linux kernel to read the last sector of a disk or partition with an odd number of sectors. Thus, those versions are able to completely forensically acquire hard disks with an odd number of sectors.

Methodology

This experiment loosely repeated the methodology used by NIST to test GNU dd and then expanded the experiment with more modern versions of the Linux kernel. The author also tested different versions of the Linux kernel using GNU md5sum as well.

Two source drives were used. First, a partition with an odd number of sectors was created on an ATA hard drive (sometimes referred to as an IDE hard drive). Although ATA hard drives with an odd number of sectors are rare, it is trivial to create a partition with an odd number of sectors. In this case, the hard drive was a Fujitsu model MPD3064AT. To create the partition, a new DOS partition table was created using fdisk and a single partition was created on the drive. The partition started at cylinder one and ended at cylinder four, making a partition of three cylinders. With each cylinder holding 20,139 sectors, this made a partition of 60,417 sectors of 512 bytes each.

The second source drive was a SCSI hard drive with an odd number of sectors. The drive was a Seagate model ST39204LC, serial number 3BV0W46F. The manufacturer's installation instructions for this drive [7] indicated that it contained 17,921,835 sectors of 512 bytes each. (It should be noted that other documents from the manufacturer [8] claim this drive contained 17,921,834 sectors.)

Four operating system environments were used in this experiment. Two of these operating systems, RedHat Linux 7.1 and FreeBSD 4.4, were installed on a hard disk. The other two operating systems came on a Knoppix boot disk[6]. Knoppix is a bootable CDROM version of the Linux operating system. This experiment used Knoppix version 3.4-2004-05-10-EN, which allows the user to select one of two kernels to be used at boot time. One kernel is version 2.4, while the other is version 2.6.

For each operating system, the computer was booted and the dd command was run in such a manner to send the data from the source drive or partition to /dev/null. The command issued was:

dd if=/dev/hda1 of=/dev/null bs=512
for the ATA drive partition and
dd if=/dev/sda of=/dev/null bs=512
for the SCSI drive.

In both cases, when the GNU dd program finished, it displayed the number of sectors processed.

Next, an MD5 hash was computed for each source drive. The GNU md5sum program uses an algorithm called MD5 to create a 128-bit output for any input. The algorithm is crafted so that even if a single bit of the input changes, the output will be radically different. It is useful for establishing whether or not two inputs are identical. In this case, the MD5 output was used to verify that the data read from the source hard drive by one version of the Linux kernel matched the data read by the other versions of the kernel. The command issued was:

md5sum /dev/hda1
for the ATA hard drive and
md5sum /dev/sda
for the SCSI hard drive.

Results

The number of sectors processed by each operating system and the MD5s for the ATA hard drive partition are shown in Figure 1. The number of sectors processed by each operating system and the MD5s for the SCSI hard drive as a whole are shown in Figure 2. With both source drives, the RedHat 7.1 and FreeBSD systems behaved identically to Lyle's earlier experiment. The RedHat Linux system missed one sector that the FreeBSD system read. The Knoppix system, when booted with a version 2.4 kernel, also missed one sector. But when booted with the version 2.6 kernel, the Knoppix system read all of the sectors successfully. Note that the MD5 hashes match for the FreeBSD and Knoppix with version 2.6 kernels. The MD5 hashes for the Redhat and Knoppix with version 2.4 kernels also match.

Operating System	Kernel Version	GNU dd version	Sectors Processed	MD5 hash
RedHat 7.1	2.4.2-2	4.0.36	60,416	7e17b8741d74dac981f81f388b862fd9
FreeBSD 4.4	(n/a)	(n/a)	60,417	6782416292bb85edfad73441e3dee03b
Knoppix 3.4	2.4.26	5.0.91	60,416	7e17b8741d74dac981f81f388b862fd9
Knoppix 3.4	2.6.5	5.0.91	60,417	6782416292bb85edfad73441e3dee03b

Figure 1 - Using dd on an ATA hard drive partition with an odd number of sectors

Operating System	Kernel Version	GNU dd version	Sectors Processed	MD5 hash
RedHat 7.1	2.4.2-2	4.0.36	17,921,834	b219fe88d89638a7c97c0aa58fa11c1d
FreeBSD 4.4	(n/a)	(n/a)	17,921,835	278afeff9cb332b6a6152633ef6512f4
Knoppix 3.4	2.4.26	5.0.91	17,921,834	b219fe88d89638a7c97c0aa58fa11c1d
Knoppix 3.4	2.6.5	5.0.91	17,921,835	278afeff9cb332b6a6152633ef6512f4

Figure 2 - Using dd on a SCSI hard drive with an odd number of sectors

Conclusion

Based on the experiment conducted, version 2.6 of Linux kernel corrects the defect found in version 2.4 and earlier versions that prevented a user land process from accessing the last sector of a hard drive or partition that has an odd number of sectors.

Disclaimer

The views expressed in this paper are the views of the author and do not necessarily represent the views of the U.S. Department of Justice or the United States.

Acknowledgements

This paper would not have been possible without the assistance of many people. Specifically, I would like to recognize:

- Joe Corrigan for conducting the initial experiment on which this project was based,
- ENS Jonathan Tighe USN for assembling the operating systems needed to run this experiment,
- Special Agents Jason Gurney and Greg Smith of the Air Force Office of Special Investigations for finding a hard disk with an odd number of sectors,

- Special Agent Barry J. Grundy of the NASA Office of the Inspector General and Brian Carrier for reviewing this paper, and
- Dr. James Lyle of NIST for his advice, support, and technical expertise.

About the author

Jesse Kornblum is the Lead Information Technology Specialist for the Computer Crime and Intellectual Property Section of the United States Department of Justice. Before coming to the Department, he served in the Air Force as an agent with the Office of Special Investigations and as an instructor at the United States Naval Academy Computer Science Department. Currently based in the Washington DC area, his research focuses on computer forensics and computer security. He can also juggle. Contact: research@jessekornblum.com; http://research.jessekornblum.com/

References

[1] National Institute of Standards and Technology, *Test Results for Disk Imaging Tools: dd GNU fileutils 4.0.36, Provided with Red Hat Linux 7.1*, 2002, http://www.ojp.usdoj.gov/nij/pubs-sum/196352.htm

[2] Lyle, James, *Notes on Odd Sized Disks and dd*, National Institute of Standards and Technology, 2002, http://www.cftt.nist.gov/Notes_on_dd_and_Odd_Sized_Disks4.doc

[3] Rubin, Paul; MacKenzie, David; Kemp, Stuart. dd is part of the GNU CoreUtils project http://www.gnu.org/software/coreutils/

[4] Brown, Michael, *Block ioctl to read/write last sector* post to the linux-kernel mailing list, Dell Computer Corporation, 2001, http://lwn.net/2001/0906/a/last-sector.php3

[5] Brouwer, Andries, *size_in_bytes* post to the linux-kernel mailing list, Centrum voor Wiskunde en Informatica, 2002, http://www.uwsg.iu.edu/hypermail/linux/kernel/0204.0/0919.html

[6] Knopper, Klaus, *Knoppix Linux Live CD*, 2004, http://www.knoppix.org/

[7] Segate, *Cheetah 18XL Installation Guide*, 2000, http://www.seagate.com/support/disc/iguides/scsi/89507b.pdf

[8] Segate, *Cheetah 18XL Family: ST318404LW/LC, ST39204LW/LC, Product Manual, Volume 1*, 2001, http://www.seagate.com/support/disc/manuals/scsi/75789506g.pdf