

Next Generation Data Forensics & Linux

Part 1: The Power of Linux

Thomas Rude, CISSP

June 2002

The field of data forensics ('computer forensics' as commonly referred to) is rapidly changing. Historically data forensics was focused on the imaging, analysis, and reporting of a stand-alone personal computer (PC) hard drive perhaps 1 gigabyte (GB) in size using DOS-based tools. However, due to a number of changes and advances in technology an evolution has begun in the field of data forensics.

The first agents of change consisted of larger hard drives. It is now common for hard drives on personal computers to be 40-60GB in size. And, in the corporate environment, it is not uncommon to have enterprise-class servers containing multiple 80GB hard drives in each. There has also been a significant increase in the number of PCs, and a noteworthy rise in the use of PCs to commit crimes or aid in criminal activities. The second agents of change include the population of non-PC devices such as handhelds, mobile cellular telephones, digital cameras, servers, etc. The third agent of change is the increase in the number of non-Windows operating systems, including both UNIX and Linux variants, MacOS, BeOS, etc.

So where do we stand today? Increasingly, forensic examiners are faced with analyzing 'non-traditional' PCs, corporate security professionals are doubling as in-house forensic examiners and incident first responders, and critical data is residing in volatile system memory. This is the 'Next Generation of Data Forensics.'

I define 'Next Generation Data Forensics' as;

The process of imaging and analyzing data stored in any electronic format, for the purpose of reporting findings in a neutral manner, with no predisposition as to guilt or innocence.

What is the Next Generation Data Forensics platform of choice? Linux. Why Linux? Linux as it stands by itself as an operating system has many features that make it a very powerful platform from which to perform data forensics work. A stock, out of the box Linux system already has built into it the ability to image, authenticate, wipe, and search media. And, these are the four critical areas in performing data forensics work. While there is no panacea for data forensics, Linux certainly is as good as it gets.

Just what makes Linux so powerful? There are a number of key functionalities within the Linux operating system environment that make it the best platform for data forensics. Among them:

- everything, including hardware, is recognized as a file
- support for numerous filesystem types
- ability to mount a file via the 'loopback driver'

- ability to analyze a live system in a safe and minimally invasive manner
- ability to redirect standard output to input, or 'chaining'
- ability to monitor and log processes and commands
- ability to review source code for most utilities
- ability to create bootable media, including floppies and compact discs

The fact that Linux sees everything as a file allows the Linux user to have some degree of control over that file. This file representation allows for ease of replication across like hardware platforms (the configuration files are clear text files). For forensic examiners, file representation allows for a couple of important functionalities; 1) a degree of security (setting access control permissions on the file) and 2) the ability to have granular control over how the operating system behaves by being able to set and manipulate these files. In terms of behavior, within Linux, the forensic examiner has the power to control how the operating system interacts with hardware devices. Why is this so critical? The number one goal of any forensic examiner is to not alter the evidence! Using Linux allows for controlled mounting of devices on a read-only basis. The forensic examiner can not only essentially tell Linux when to touch devices and when not to touch devices, but also how to touch the devices as well as interact with the devices once touched.

Another very powerful feature of Linux is the number of filesystems it supports. Most recent Linux distributions support numerous filesystems, including: ext2, ext3, ADFS, FFS, HPFS, befs, FAT, VFAT, RamFS, ISO9660, NTFS, UDF, UFS, NFS, NCP, among others. Having support for so many different filesystem types means the forensic examiner does not have to rely on a software vendor to write a driver for that filesystem. The driver is already there. The advantage is that one Linux platform can use these filesystem drivers to interpret the data from many different operating systems and their respective filesystems. No more booting different operating systems just so that the logical data can be interpreted! It is also very important to note that this feature is critical to those examiners who have to drop their attack box in a foreign network to perform live analysis. For example, Linux would be a powerful resource to have available when responding to a breach in a server farm environment running a UNIX variant and NFS.

The loopback device is recognized as a special block device in Linux. When used, it allows for a file to be mapped onto a virtual block device. Block devices are devices that deal in blocks of data (hard drives, etc.). So how can the loopback device assist the forensic examiner? Let's say for example an examiner creates an image of a partition using the 'dd' (data dump) command. Now there exists a single file (created by the 'dd' command) that contains the entire logical partition. Using a hex editor to view the contents of this file could be done, but would prove to be trying and time consuming. Instead, the forensic examiner could take this single image file and mount it using the loopback device. Once mounted, the examiner can now navigate and browse the image file just as if it were the original physical hard drive with the logical partition available for analysis. Let's also take a look at another example of when the loopback device will assist in data forensics. Let's say Examiner A creates a dd image of an entire physical disk that contains multiple partitions. Examiner A then turns over a copy of this evidentiary image to Examiner B. Examiner B has no knowledge of what operating system(s) or filesystem(s) reside within the image file, thereby making the analysis process a bit tougher. There is a way to circumvent this lack of knowledge. The examiner can use the loopback device to mount this single image file containing an entire hard drive, passing options to the mount command that will allow for the mounting of this single file as a block device. Once mounted, the examiner can view and navigate just as if the physical hard

drive were attached to the analysis platform.

The power of Linux is not limited to post mortem analysis of a dead system. In addition to this type of analysis, Linux can also be used to analyze an up and running live system. The features of Linux allow for this live analysis to be conducted in a safe and minimally invasive manner. Again, remembering that a primary goal is to not alter or erase potential evidence, then a method for conducting analysis on a live system needs to be proven to not only minimize any stepping on the system, but it also must be accurate and the data gleaned must be trusted. In other words, the environment of a compromised system cannot be trusted, or trusted only minimally. With the design of the Linux operating system a forensic examiner can create safe media from which to conduct the live analysis. Examples of safe media include floppy diskettes and CD-ROMs. The content of these may vary from examiner to examiner, but the provision remains the same – statically linked binaries from a trusted source. No matter what tools are selected for inclusion, the examiner has a trusted toolbox from which to work.

Linux took a number of functionalities from its big brother, UNIX. One of which really enables the forensic examiner to work in an efficient manner. The ability to redirect output from one command and input it to another command is a very powerful feature of Linux. Instead of having to issue one command and wait for it to complete before issuing a second command, commands may be issued in such a way that they are 'chained' together and they may be issued in a manner such that the result of the first command is piped into the second command. Again, there are options that can be passed when chaining or redirecting so as to insure that the desired outcome (integrity, accuracy, etc.) is achieved. A simple example of this would be the following command;

```
md5sum /dev/hdb > hash_value.txt ; dd if=/dev/hdb of=/dd_image
```

In this example a hash value of the physical hard disk HDB is calculated by md5sum, the result of which is written to the file 'hash_value.txt'. Upon completion of this authentication the command 'dd' is issued to create an image of the physical hard disk HDB and resulting image file is 'dd_image'. Notice that the semi-colon was used to chain these two commands together such that the completion of the first signals the start of the second. Also notice that the output of md5sum is redirected from standard out (monitor) to 'hash_value.txt'. Imagination and technical competence become the only limitations a forensic examiner has here.

The design and commands of the Linux environment make it possible to log and monitor just about anything that is happening on the system. The value to the forensic examiner is a trail of what happened when, and by whom or what. Not only is this useful for recording steps taken during the post mortem analysis of a dead system, but perhaps even more useful and important is the ability to log and monitor on a live system (especially if trusted binaries are used). The analysis of a live system in a foreign environment does not have the luxuries afforded to it as does a post mortem analysis at the forensic examiner's lab. However, by using the trusted binaries (or even the non-trusted compromised system binaries if necessary) the examiner has the power to log every command issued and record that log to a file. This log file may prove invaluable in terms of non-repudiation as well as an excellent reminder months later of what was done and when it was done.

As part of the Open Source environment, Linux and many of the tools used by forensic examiners have source code available for review. The ability to review the source code proves to be beneficial to the forensic examiner in many aspects. Knowing what a command or tool is really doing 'behind the scenes' is invaluable

to an examiner. In terms of being able to defend what was actually done and perhaps even how it was done, the ability to review the source code is essential. If a technical, low level discussion is required, reviewing the code can be an excellent source of information. Not only can code review assist the examiner and improve the examiner's competence, but having the code available makes it easier to modify or customize the code.

No discussion on the value of Linux as a data forensics platform can be complete until the topic of bootable media and/or custom distributions is covered. The design of the Linux operating system is such that a single floppy diskette may have a bootable and fully functional operating system on it. And since the Linux operating system footprint can be very small in size, the ability to include useful utilities and commands is available. A forensic examiner can have the tools required for imaging, authenticating, and searching all on a single, bootable Linux floppy diskette. But, even better, on a standard CD-ROM, the examiner now has 600+ megabytes (MB) available to include the operating system, tools, and utilities. These safe media disks can either be bootable or non-bootable. Perhaps it's best to have two of each!

All of the aforementioned features of Linux make it a very powerful platform from which to conduct the data forensics process. These features, in and of themselves, empower the forensic examiner to a high degree. And it is because of these features that numerous tools have been developed that help aid in the processing of data forensics. Some of these tools have been developed specifically with the forensic examiner in mind while others were developed for network and systems administrators. No matter the intended user, these tools build upon the features and functionalities of Linux. Typically these tools aid in a number of ways, from automating manual tasks to reporting findings in a more user friendly format, to taking the power of the command line to a X-Window System-based graphical user interface (GUI). Part 2 of this series will focus on the tools available, their impact, and their use with respect to data forensics.