

Chapter IV

Digital Forensics Tools: The Next Generation

Golden G. Richard III, University of New Orleans, USA

Vassil Roussev, University of New Orleans, USA

Abstract

Digital forensics investigators have access to a wide variety of tools, both commercial and open source, which assist in the preservation and analysis of digital evidence. Unfortunately, most current digital forensics tools fall short in several ways. First, they are unable to cope with the ever-increasing storage capacity of target devices. As these storage capacities creep into hundreds of gigabytes or terabytes, the traditional approach of utilizing a single workstation to perform a digital forensics investigation against a single evidence source (e.g., a hard drive) will become completely intractable. Further, huge targets will require more sophisticated analysis techniques, such as automated categorization of images. We believe that the next generation of digital forensics tools will employ high-performance computing, more sophisticated evidence discovery and analysis techniques, and better collaborative functions to allow digital forensics investigators to perform investigations much more efficiently than they do today. This chapter examines the next generation of digital forensics tools.

Introduction

A wide variety of digital forensics tools, both commercial and open source, are currently available to digital forensics investigators. These tools, to varying degrees, provide levels of abstraction that allow investigators to safely make copies of digital evidence

and perform routine investigations, without becoming overwhelmed by low level details, such as physical disk organization or the specific structure of complicated file types, like the Windows registry. Many existing tools provide an intuitive user interface that turns an investigation into something resembling a structured process, rather than an arcane craft.

Unfortunately, the current generation of digital forensics tools falls short in several ways. First, massive increases in storage capacity for target devices are on the horizon. The traditional approach of utilizing a single workstation to perform a digital forensics investigation against a single evidence source (e.g., a hard drive) will become completely intractable as storage capacities of hundreds of gigabytes or terabytes are seen more often in the lab. Furthermore, even if traditional investigative steps such as keyword searches or image thumbnail generation can be sped up to meet the challenge of huge data sets, much more sophisticated investigative techniques will still be needed. For example, while manually poring over a set of thousands (or even tens of thousands) of thumbnails to discover target images may be possible, what will an investigator do when faced with hundreds of thousands of images? Or millions?

The next generation of digital forensics tools will employ high performance computing, more sophisticated data analysis techniques, and better collaborative functions to allow digital forensics investigators to perform investigations much more efficiently and to meet the challenges of massive data sets. In this chapter, we examine some of the technical issues in next generation tools and discuss ongoing research that seeks to address them.

Challenges

To see the challenges faced by the next generation of digital forensics tools, we examine the looming problems of scale that will soon overwhelm current generation tools. The primary challenges are fueled by fundamental trends in computing and communication technologies that will persist for the foreseeable future. Storage capacity and bandwidth available to consumers are growing extremely rapidly, while unit prices are dropping dramatically. Coupled with the consumer's urge to have everything online, where music collections, movies, and photographs will increasingly be stored solely in digital form, these trends will result in even consumer-grade computers having huge amounts of storage. From a forensics perspective, this translates into rapid growth of the number and size of potential investigative targets. To be ready, forensic professionals need to scale up both their machine and human resources accordingly.

Currently, most digital forensic applications are developed for a high-end, single or dual-CPU workstation that performs queries against a set of target media. In our experience, this approach is already very time-consuming, even for targets of modest size. More importantly, fundamental trends in hardware dictate that this single workstation approach will hit an insurmountable performance wall very soon. Patterson (2004) performed a quantitative survey of long-term trends in hardware with respect to capacity, bandwidth, and latency. From a forensics perspective, the most consequential result is the observed divergence between capacity growth and improvements in latency. Spe-

cifically, over the last 10 years, for representative “high performance” hard disk drives, the capacity has grown 17 times (from 4.3 to 73.4 GB), while average latency (disk seek time) has improved only 2.2 times (from 12.7 to 5.7 ms). Similarly, the gap between capacity and transfer rate has also grown as transfer rate (throughput) has improved only 9.6 times (from 9 to 86 MB/s). In practical terms, the gap is even bigger among high-capacity (250GB+) drives targeted at the mass retail market. These are typically EIDE/ATA drives that are optimized for capacity and cost, with throughput and latency being somewhat less important.

Since most current digital forensics operations, such as computing cryptographic hashes, thumbnail generation, file carving, and string searches, are I/O-bound, the performance of existing investigative tools will become completely unacceptable as the size of the problem (determined by capacity) grows significantly faster than the ability to process it (determined by drive latency and transfer rate limitations). We refer to the ability to scale up machine resources to match the growth of the forensic targets as *machine scalability*.

A generally overlooked side of the scalability problem, which we refer to as *human scalability*, is the ability to make efficient use of human resources in a digital forensics investigation. This includes the presence of more advanced processing capabilities to relieve experts from routine work (e.g., searching for contraband images) as well as collaborative support. Collaborative support allows multiple experts to efficiently work together on a case.

An alternative view of scalability is to consider turnaround time of time-sensitive digital forensic investigations. For example, consider a situation where law enforcement officers have seized a computer belonging to a kidnapping suspect. In this situation, it is *critical* that investigators be able to concentrate all available machine/human resources (perhaps in an ad-hoc manner) and thoroughly examine the available information for clues as rapidly as possible. Turnaround of minutes or hours is needed, rather than days or weeks.

For all practical purposes, current tools do not deal with scalability issues of the kind described above. Therefore, in the following sections, we discuss in more detail both the machine and human aspects of the scalability problem and present some approaches to address them.

Machine Scalability

At a high level, the technical aspects of the digital forensic process can be described as follows: for each file in a given file system, perform a number of type-specific operations—indexing, keyword searches, thumbnail generation, and others. Digital evidence such as deleted files, file slack, directory structures, registries, and other operating system structures can be represented as special file types, so the model applies to these types of evidence as well. To be credible, an investigator must usually thoroughly examine the content of the entire forensic target. Even in cases where a partial examination is acceptable, a substantial amount of data must be processed. Thus, the turnaround time of a forensic inquiry is inherently limited by disk transfer rate and seek time.

Current tools, such as the Forensics Toolkit (FTK) from AccessData Corp., attempt to reduce the need to read an entire forensics image repeatedly (e.g., for each search operation) by performing an initial preprocessing step that builds up some index structures to speed up keyword searches, disk carving, and to provide file categorization. While this technique is effective in many scenarios, it is limited by the computational resources available on a single workstation. First, it may take several days just to perform the preprocessing step. Second, the system indexes only strings that it judges to be of use in the investigation: for example, character sequences that appear to be similar to English words and those that are useful for file carving. Regular expression searches, as well as simple searches for character sequences that are not in the index, such as words in foreign languages with different encoding, still require an exhaustive examination of the entire target image. On targets of hundreds of gigabytes or terabytes, investigators may (necessarily) be disinclined to perform searches that may take days of execution time, particularly as caseloads grow. Finally, the index structure of a large target will also become large, which will prevent it from being kept in main memory.

Generally, there are two possible approaches to improve machine scalability—improve the efficiency of the algorithms and their implementations to get more from the current hardware platforms or enable the use of more machine resources in a distributed fashion. These two approaches are to a great extent complimentary; however, the former is likely to yield only incremental improvements in performance, whereas the latter has the potential to bridge the hardware performance gaps discussed earlier. The reason for this is that most investigative techniques offered by the current generation of digital forensics tools are I/O-bound. More sophisticated investigative techniques, such as image classification, suffer from both the I/O bottleneck (because images must be completely retrieved to be classified) *and* require substantial CPU resources.

In summary, while any kind of digital forensics analysis is inherently I/O-constrained because of the need to process vast amounts of data, it can also become CPU-constrained if more sophisticated analytical techniques are used. A distributed solution can address both the I/O and the CPU constraints. For example, a 64-node Beowulf cluster with 2GB of RAM per node can comfortably cache over 100GB of data in main memory. Using such a system, the cost of the I/O transfer of a large forensic image can be paid once and any subsequent I/O can be performed at a fraction of the cost. Taking the idea a step further, the data cached by each node can be made persistent so that if the system needs to shutdown and restart, each node need only autonomously read in its part of the data from a local disk. At the same time, having multiple CPUs performing the CPU-intensive operations obviously has the potential to dramatically improve execution time. Therefore, in the following section, the focus of the discussion is on the application of distributed computing techniques in a digital forensics environment.

Distributed Computing and Digital Forensics

Most digital forensics operations are naturally file-centric with very few (if any) dependencies among the processing of different files. Thus, choosing an individual file as the primary distribution unit minimizes synchronization and communication among the nodes of the cluster. Consequently, the first essential step in employing distributed

computing is to distribute the files comprising the digital evidence over a compute cluster.

From a caching perspective, maximizing speedup is relatively straightforward—files should be spread such that as many of them as possible are kept in RAM during processing. Large files that are much bigger than the available physical memory on any given machine may have to be split into pieces and/or processed separately. It is desirable, but not crucial, that there be enough physical memory to cache all useful files during processing. But RAM “overloading” will automatically be handled by the host virtual memory system. Although no experimental results have been published, common experience from general operating system usage suggests that, depending on access patterns, overloading by as much as 50% can have only modest impact on performance, and as much as 100% may be tolerable.

Maximizing CPU utilization is a bit more complicated. One approach is to scatter the files of a particular type evenly across the processing nodes. The rationale is that whenever an operation is issued, for example, a regular expression search, all nodes will have a similar amount of work to complete and, therefore, CPU utilization will be maximized. However, more sophisticated processing that attempts to correlate different objects (such as the image classification technique discussed later) may be hampered by this file distribution pattern, increasing the need for network communication. In this case, concentrating the files in fewer nodes and crafting a suitable communication pattern may yield better results.

Another twist is the recent trend toward routine use of symmetric multi-processor (SMP) systems, especially in high performance compute clusters. In an SMP, all CPUs have uniform access to a shared memory pool and often have dedicated high-speed communication among the processors. Clearly, to optimize performance, such architectural features must be taken into consideration during the distribution and processing phases.

Distributed digital forensics tools are still in their infancy but even preliminary results from research prototypes clearly demonstrate the benefits of the approach. *DELV* (Distributed Environment for Large-scale investigations) provides a look at how distributed systems can be applied to digital forensics (Roussev & Richard, 2004). An investigator controls the investigation on a single workstation through a GUI similar to those provided by other forensic tools in common use. Behind the scenes, however, digital forensics operations are farmed out to nodes in a commodity *Beowulf* cluster and the returned results are aggregated and dynamically presented to the user as soon as they become available. Thus, to perform a complicated regular expression search against a large target, for example, the investigator enters a single expression and the search is performed in parallel across all (or some subset of) the cached evidence. As hits accumulate, they are displayed for the user.

There are three notable differences in the user experience between *DELV* and most traditional, single-machine digital forensics tools. First, the system does not perform any preprocessing—it simply loads the forensic image and is ready to perform queries. The system supports two different modes to load target images. The first is “cache” mode, in which a central coordinator node reads the entire image and distributes data over the network to compute slaves. In the other “load” mode, the coordinator instructs the slaves to individually load certain data from the target image, which is on a shared fileserver.

Preliminary experiments have shown that the concurrent loading provided by “load” mode was much better able to utilize the read throughput of a high performance RAID storage, by more than 30% in some cases. Nodes can use their local disk to cache their part of the evidence so subsequent loads of the image take only a fraction of the original time.

Another difference is that since all work is performed remotely, the investigator’s machine remains responsive and available to do follow-up work on the partial results (e.g., open a matching file) as soon as they become available. It is also possible to start new queries, for example, text searches, while previous ones are still running, with little noticeable change in the overall performance. This is due to the fact that many operations, such as text searches, are I/O-bound. Once the I/O bottleneck is overcome through caching, the CPUs can easily handle simultaneous queries. More generally, it is reasonable to expect the execution time of overlapping I/O-bound operations to be very close to that of a single query.

The final difference is that investigative operations execute in a fraction of the time required on a single workstation. Specifically, the 8-node experiments in (Roussev & Richard, 2004) point to a super-linear speedup for I/O-bound forensics operations. The speedup in this case is likely to be a constant factor that is not related to the concurrency factor (number of nodes) but reflects the time savings from not accessing the disk. Nonetheless, the gap between cluster and single workstation performance grows as a function of the target size. This occurs because as the resource mismatch between a single workstation and the target processing requirements grows, other adverse side effects such as virtual memory system thrashing and competition for RAM resources between index structures and evidence seriously degrades performance. For CPU-bound operations, such as detection of steganography, the observed *DELV* speedup is approximately equal to the concurrency factor.

Although these results are still early work, they provide some food for thought in improving the processing model of digital forensics tools. One important issue is to improve investigation turnaround time. For example, if the complete target can be kept cached in RAM, costly preprocessing (such as string indexing), designed to speedup I/O-bound operations such as string searches, can be completely eliminated in favor of an on-demand distributed execution of the operation. Another attractive possibility is to perform the preprocessing step in parallel on the cluster and then use the results on local workstations. This may not be possible if the specific processing needed is only available from a proprietary software package, such as *FTK*. However, it might still be possible to pool the RAM resources of the cluster and create a distributed RAM drive. Assuming a fast enough network (e.g., gigabit or better), such a network “drive” should outperform a local hard disk when a significant fraction of the disk operations are non-sequential.

Looking forward, distributed computing also allows the sophistication of investigative operations to be improved substantially. For example, automated reassembly of image fragments (Shanmugasundaram, 2003) and analysis of digital images to determine if they have been tampered with or were computer-generated (Farid & Lyu, 2003), watermark detection (Chandramouli & Memon, 2003), automatic detection of steganography (Chandramouli, Kharrazzi, & Memon, 2004), and correlation and attribution (de Vel, Anderson, Corney, & Mohay, 2001; Novak, Raghavan, & Tomkins, 2004) of documents

all have significant computational requirements and will be made practical by the application of high-performance computing.

Some digital forensics operations straddle the machine vs. human scalability line. Sophisticated image analysis is one example, where deeper analysis of images can save a significant amount of human effort, but the analysis may only be feasible if sufficient computational resources can be applied. Content-based image analysis, which fits into this category, will be discussed in a subsequent section.

On-the-Spot and “Live” Digital Forensics

Another approach to improving machine scalability is to do a better job with preliminary identification of evidence. Currently, the best practical solution in large-scale investigations is to either seize all sources of evidence or use a portable high performance storage system to obtain a copy of any potential evidence. There are several reasons that this approach is problematic. The first has already been discussed—as forensics targets grow in size, which they are doing already at an overwhelming pace—insurmountable logistical problems will arise in the collection, preservation, and analysis steps of an investigation. In some cases, a forensic target may be a currently unidentified machine (or machines) in a large network, for example, in a computer lab at a library. In other cases, the forensic target might be a huge fileserver, whose operation is critical for the well-being of a company. Performing an imaging operation on every machine in a large laboratory setting will be a very daunting task, as will imaging a multi-terabyte fileserver. Even if logistical problems with the imaging process are overcome, a huge interruption of service is necessary during a traditional imaging operation, during which normal operation of the computer systems is impossible. Finally, analyzing the drives of a large group of machines (or of a terabyte fileserver) will consume considerable resources.

A more efficient solution is to perform a safe screening of the target systems and take only the relevant data and systems to the lab. Furthermore, such screening can be performed using the local computational and communication resources of the targets. A straightforward solution which overcomes some (but not all) of the logistical problems described above is creation of better imaging tools, where files that are not interesting (e.g., operating systems files or file types irrelevant to an investigation) are not included in the captured image. In many cases, however, the number of files that might be excluded may be rather small, in comparison to the size of the entire target. Thus, other approaches should be explored, in addition to creating better drive imaging tools.

The Bluepipe architecture (Gao, Richard, & Roussev, 2004) permits an on-the-spot investigator to perform simple queries and to capture and preserve digital evidence, using only a small amount of hardware (e.g., a PDA or laptop). Bluepipe uses a client/server architecture, with a server running on the target machine and one or more Bluepipe clients controlling the investigation. The communication between client and server is via a SOAP-based protocol. Bluepipe clients may also serve as proxies, to allow a remote investigator to participate in a collaborative fashion.

To begin an inquiry, an investigator performs several steps: she plugs in USB dongles to enable wireless communication with the target computers, boots the target computers

using Bluepipe boot CDs, and launches the Bluepipe client application on her PDA or laptop. The Bluepipe boot CD invokes the server-side Bluepipe application, initializes the connection between client and server, and exposes the secondary storage devices of the target to the Bluepipe server application. The investigator then uses the client GUI on the PDA (or laptop) to issue queries and receive results. All processing on the target side consists of collections of read-only operations—called Bluepipe patterns—against the secondary storage on the target machine. An audit log tracks all operations performed on the target; this log is transmitted to the client at the end of the inquiry. Because some investigatory operations are expected to complete quickly and some require substantial processing time, Bluepipe supports both synchronous and asynchronous communication.

A Bluepipe investigation consists of execution of a number of Bluepipe patterns. A Bluepipe pattern is an XML document describing a set of related operations to be executed on the target machine, combined with some additional parameters that govern priority and frequency of progress updates. The goal of a pattern might be to determine if a particular application is installed on the target, to extract a system timeline, or to perform keyword searches for certain credit card numbers. All Bluepipe patterns preserve the state of secondary storage on the target machine. Supported pattern operations include checking for existence of files with specific names or hash values, searching files for keywords, retrieving files, and generating directory and partition table listings. Bluepipe patterns are stored on the client and transmitted to the Bluepipe server for execution as they are selected by the investigator. Results of the pattern execution are then transmitted back to the client.

A few simple examples illustrate the use of Bluepipe patterns to perform preliminary analysis of a target machine. The following pattern was used to obtain a partition table listing of a target with a single IDE hard drive:

```
<BLUEPIPENAME="partitions">
  <!-- get a lot of drive/partition info-->
  <LISTPARTITIONSLOCAL="drives.txt"
  GENHASHES=TRUE/>
</BLUEPIPE>
```

The result of executing this pattern, a text file named “drives.txt”, illustrates that the target machine’s single hard drive contains five partitions with at least two operating systems installed:

```
hda
Model Number: IC25T060ATCS05-0.
Serial Number: CSL800D8G3GNSA
device size with M = 1024*1024: 57231 Mbytes
```


Partition table:

Disk /dev/hda: 240 heads, 63 sectors, 7752 cylinders

Units = cylinders of 15120 * 512 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	1	6173	46667848+	7	HPFS/NTFS	
/dev/hda2	7573	7752	1360800	1c	Hidden Win95 FAT32 (LBA)	
/dev/hda3	*	6174	7364	9003960	83	Linux
/dev/hda4	7365	7572	1572480	f	Win95 Ext'd (LBA)	
/dev/hda5	7365	7572	1572448+	82	Linux swap	

MD5 hash for drive: 463e65ec8d9f51bdd17c0347243f467b

The next pattern, named “findcacti”, searches for pictures of cacti using a hash dictionary. A single target directory is specified, “/pics”, which is searched recursively. Files that match are retrieved and stored on the client in a directory named “cactus”. No file size restrictions are imposed. The %s and %h placeholders in the message will be replaced by the filename and hash value of each matching file.

```
<BLUEPIPENAME="findcacti">
<!-- find illegal cacti pics using MD5 hash dictionary -->
<DIR TARGET="/pics/" />
<FINDFILE
USEHASHES=TRUE
LOCALDIR="cactus"
RECURSIVE=TRUE
RETRIEVE=TRUE
MSG="Found cactus %s with hash %h ">
<FILE ID=3d1e79d11443498df78a1981652be454/>
<FILE ID=6f5cd6182125fc4b9445aad18f412128/>
<FILE ID=7de79a1ed753ac2980ee2f8e7afa5005/>
<FILE ID=ab348734f7347a8a054aa2c774f7aae6/>
<FILE ID=b57af575deef030baa709f5bf32ac1ed/>
<FILE ID=7074c76fada0b4b419287ee28d705787/>
<FILE ID=9de757840cc33d807307e1278f901d3a/>
```

```

<FILE ID=b12fcf4144dc88cdb2927e91617842b0/>
<FILE ID=e7183e5eec7d186f7b5d0ce38e7eaad/>
<FILE ID=808bac4a404911bf2facaa911651e051/>
<FILE ID=fffbf594bbae2b3dd6af84e1af4be79c/>
<FILE ID=b9776d04e384a10aef6d1c8258fdf054/>
</FINDFILE>
</BLUEPIPE>

```

The result of executing this pattern on a target appears below. Notice that the DSC00051 and bcactus5 image files have identical content:

```

Beginning execution for pattern "findcacti".
DIR cmd, added "/pics".
FINDFILE cmd.
Found cactus/pics/BBQ-5-27-2001/DSC00008A.JPG with hash
6f5cd6182125fc4b9445aad18f412128
Found cactus/pics/BBQ-5-27-2001/DSC00009A.JPG with hash
7de79a1ed753ac2980ee2f8e7afa5005.
Found cactus/pics/CACTUS_ANNA/DSC00051.JPG with hash
3d1e79d11443498df78a1981652be454.
Found cactus/pics/GARDEN2002/bcactus5.JPG with hash
3d1e79d11443498df78a1981652be454.
Pattern processing completed.
Sending pattern log. Remote filename is "findcacti.LOG".

```

Ultimately, tools like Bluepipe don't attempt to replace traditional methods in digital forensics—instead, they improve the triage process and also improve the availability of digital forensics investigators. Another type of tool, which also improves triage but operates on live machines, is described below.

An interesting trend in next-generation digital forensics is “live” forensics investigation—analysis of machines that are allowed to remain in operation as they are examined. The idea is appealing, particularly for investigation of mission-critical machines, which would suffer a substantial downtime during a typical “dead” analysis. The mobile forensic platform (Adelstein, 2003), now called the OnLine Digital Forensic Suite in its commercial incarnation, allows live investigation of computer systems, permitting investigators to obtain evidence and perform a thorough investigation remotely. The researchers observe, quite correctly, that in large computer networks, unauthorized activity can have devastating consequences and must be dealt with very quickly. Unfortunately, most organizations simply do not have the staff to examine each local

network potentially involved in an attack. In addition, in any geographically dispersed organization, the less time the investigators spend traveling, the more time they have to investigate the incident. This applies to networks that span a few buildings, let alone a city or a country. The MFP is a network appliance, deployed on an organization's local network, which exposes a secure, Web-based investigative interface to an organization's computers. The machines may be investigated while they perform their usual functions, without raising the suspicion that they are under investigation.

A live investigation using the MFP will involve collecting evidence from one or more targets. The MFP organizes an investigative effort into inquiries, each of which represents an investigator's effort to collect data from a target. During a particular inquiry an investigator may collect a machine's state, including running processes, a list of who is currently logged in, and networking information such as currently executing servers and which ports they are listening on. During the inquiry, the investigator may also capture memory dumps of physical memory and running processes, examine the registry (for Windows) and copy files from the target to the MFP network appliance. Any analysis is then performed on data acquired during a particular inquiry—should the investigator wish to snapshot the machine's state again, an additional inquiry is created. Time-consuming operations, such as capturing the physical memory of the target or imaging the entire disk, run as background threads in the MFP and do not tie up the user interface. This design choice should be made in all future digital forensics tools, as we point out in a following section.

One important difference between a traditional “dead” digital forensics investigation—where a machine is seized, its drives imaged, and analysis performed on these copies—and a “live” investigation, using the MFP, is that the investigator is not playing an adversarial role. The MFP requires administrative privileges on the machine under investigation and uses the operating system and hardware resources of the target. As such, it may not be possible to investigate machines whose operating systems have been completely compromised, through the installation of kernel-level rootkits, or machines whose administrator account passwords have been (maliciously) changed. For these kinds of situations, a traditional “dead” analysis is likely required, though all contextual evidence, such as what processes were running, who was connected to the machine, and what information is resident only in memory, will be lost when the machine is taken down.

Human Scalability

Improving human scalability means making better use of an investigator's time, automating tasks that are routine or tedious, and saving brainpower for tasks that require human intelligence. One benefit of applying high-performance computing to digital forensics investigations is that the abundance of computational resources allows the creation of tools that are much more responsive to an investigator. That is, investigators might continue to work on other aspects of a case while searches and other processing occurs in the background. Highly responsive, multithreaded GUIs are a requirement for next-generation digital forensics tools.

Another benefit is that high-performance computing allows substantially more sophisticated investigative techniques to be supported. For example, the average computer user will likely have a substantial collection of the multimedia objects, such as images, audio, and video files. Existing tools provide almost no automation for investigation of multimedia—essentially, an investigator must examine each file in turn. There are a number of digital signal processing techniques that can be employed to speed up the analysis of multimedia. However, such approaches require substantially more computational resources than a single- or dual-CPU system can offer, so high performance computing is a de facto prerequisite for the practical use of such techniques. The next section discusses early research efforts aimed at automating the processing of multimedia evidence as well as some ideas on the kind of support that can be expected in the coming years.

Automated Image Analysis

Digital forensic investigators are often faced with the task of manually examining a large number of digital pictures in order to identify potential evidence. The task can be especially daunting and time-consuming if the target of the investigation is very broad, such as a Web hosting service. Current forensic tools are woefully inadequate in facilitating this process and their support is largely confined to generating pages of thumbnail images and identifying known files through cryptographic hashes. Several more sophisticated techniques for processing images are discussed below.

Content-based image retrieval (CBIR) techniques (Chen, Roussev, Richard, & Gao, 2005) have the potential to dramatically improve the performance of image-based searches in at least two common scenarios—queries for contraband images and queries for images related to some known images (e.g., a picture of a particular person). A CBIR system works by extracting and storing a set of image features—essentially, mathematical properties of an image—for each target image. One mathematical approach to extract these features is described in Chen et al. (2005); the interested reader is referred there for the details. Intuitively, the feature set can be thought of as a form of “fingerprint” of the image and can be used later to automatically identify the original image and some versions of it. Based on the feature information of a target set of images, the system builds a database that can later be queried by submitting images or feature sets. The result of a query is a ranking of the images in the database with the one most similar to the query at the top.

To use CBIR for contraband discovery, the feature set database is updated by various law enforcement agencies with the feature sets of discovered contraband images. Thus, all images on an investigative target can be automatically compared to the ones in the features database. To use CBIR for image queries, the system first builds a database from all the images on the target and then allows the investigator to submit image queries that rank target images by similarity.

The CBIR approach has several properties that make it particularly suitable for digital forensics purposes:

- **Source independence:** The original images are *not* stored and it is *not* possible to recover them in any form from the stored feature data. This is particularly important in storing information about contraband images, since direct storage of the images themselves is often illegal. Even if legality is not an issue, the use of features instead of originals essentially eliminates the security and public relations risks associated with maintaining the database.
- **Scalability:** The storage requirements for the extracted information are a small fraction of those of the original image. For high resolution images, less than one percent is typical. This allows the resulting system to scale much better than one based on direct image-to-image comparison and will certainly offer better response time for database queries.
- **Stability:** In addition to discovering exact copies of query images, a CBIR repository system has the added advantage that it can readily identify common image variations. In Chen et al. (2005), the ability of a CBIR system to match a transformed image to its original was evaluated. The system was over 99 percent accurate in identifying a target image, even after substantial reductions in size or quality. 90-degree rotations and mirroring transformations had a similar effect on the system's effectiveness. In contrast, most existing image query techniques are based solely on cryptographic hashes. This type of matching is very fragile, because only identical files can be discovered. Finally, the stability of CBIR methods further improves the scalability of the system as only a single feature set needs to be stored for a group of derived images.

Image clustering can be built on top of the CBIR approach and seeks to help an investigator by automatically separating target images into clusters of similar images. The idea is to enable the investigator to quickly get an idea of the image content of a large target by looking at a few representative images from each cluster. The flip side of this kind of analysis is to find “anomalies” in the image distribution. For example, it may of interest to flag images that are stored in the same directory, but which have very different content. Obviously, image clustering will not replace human judgment in the forensic process, but has the potential to drastically reduce the time required to find evidence of interest.

Streaming Media Analysis

Looking forward, ordinary users will increasingly have large libraries of streaming multimedia content. Today, there are practically no tools for automating the examination of such evidence, beyond extraction and searching of any embedded textual information. Part of the problem is that the single-CPU machine is already pushed to the limit and automated (CPU-hungry) analysis is simply not practical. However, a distributed platform offers enough power to tackle the problem. Some ideas for research in this area include:

- **Automated video summarization:** The forensic system can be tasked to extract a series of “important” images that characterize the video stream to be shown to the investigator. Image processing techniques, such as image clustering or feature identification, can be then applied to the individual images.
- **Voice identification/characterization:** Voice analysis tools have been used for a while but are generally not available for routine inquiries. Potential applications include finding occurrences of a specific person’s voice in an audio file or identification of the voices of children. The idea is automate these processes and enable their use on large-scale targets.
- **Searchable multimedia:** The basic idea is to combine automated video summarization with speech-to-text conversion to produce an HTML-like summary that can be browsed and searched with conventional tools.

Multi-User Tools

Another side of human scalability is efficiently pooling the knowledge and expertise of a team of investigators. There are at least two kinds of support that teams need—real-time and long-term. Real-time support is needed to allow teamwork on the same case, so that investigators can see each other’s actions and results and coordinate on different aspects of a case. The same technology can also be used for training purposes, allowing an inexperienced investigator to observe the approaches taken by more experienced investigators.

Real-time collaboration support becomes particularly relevant if the team has access to a high performance compute cluster. On the one hand, the distribution of data and computation *enables* the parallel execution of multiple operations (perhaps submitted by different team members). At the same time, the cluster becomes a valuable resource that virtually *requires* the ability to dynamically share it across teams/cases for proper utilization. Providing real-time collaboration support will require more sophisticated user interfaces, to control the collaboration, additional security mechanisms beyond those provided in typical single-user tools, and more sophisticated concurrency control, to protect the integrity of a digital forensics investigation. Real-time collaboration support is currently being implemented as part of the work described in Roussev and Richard (2004) and Gao et al. (2004).

Long-term collaboration support refers to the ability of the digital forensics infrastructure to efficiently store and present the technical knowledge accumulated through the processing of different cases. Digital forensics knowledge bases are an obvious choice for supporting the exchange of forensic expertise within the lab and across the digital forensics community. In general, even though a knowledge base may present a unified interface to access the “lessons learned”, care must be taken because internal and external sources may have different sharing restrictions, trustworthiness, and structure. Internal sources are presumably based on existing cases and an appropriate level of confidentiality must be maintained. Alternatively, lessons could be anonymized.

The work described in (Mandachela, 2005), called a digital forensics repository (DFR), is an early attempt to address the needs of long-term collaboration through a specialized knowledge base. The central idea, borrowed from (Harrison, 2002), is to build a repository of lessons. A lesson is any technical article that describes a procedure/method for solving a particular forensic problem, such as imaging a specific type of device. Lessons may be created from reports generated by popular digital forensics suites, imported from the Web, or created manually. The system also supports RSS feeds to distribute new lessons and features such as a “lesson of the day”.

Conclusion

The technical challenges facing next generation digital forensics tools are dominated by issues of scale. Current single-CPU systems are quickly approaching a point where their poor performance will make them unusable, due to a fundamental imbalance between the resources needed to process the target and the resources available on a single forensics workstation. The only way to address this imbalance is to base the next generation of digital forensics tools on a high performance computing platform, while simultaneously trying to improve the user experience of investigators using the tools and improving the evidence acquisition process. While some problems with current tools—such as lack of multithreading, which often results in unresponsive user interfaces during intensive tasks—are easily corrected with incremental improvements to the applications, new approaches are required to deal with these issues of scale. In addition to sophisticated evidence caching schemes and the use of more CPUs, better collaborative capabilities are also needed, to allow investigators to work together on difficult cases.

Early experimental results in distributed digital forensics confirm that this approach is indeed a practical one, in many cases yielding speedups that well exceed the concurrency factor. A distributed computing approach also allows interactivity to be improved and will enable deployment of sophisticated methods for multimedia processing into next generation tools. For example, next generation tools should offer investigators far more powerful facilities for images and video than simple thumbnailing, including automatic categorization of images, image searches which are immune to typical image transformations, and summarization and searching for video files. Distributed computing will make implementation of these facilities possible—a resource-starved, single CPU workstation simply isn’t up to the task.

Some new tools are also becoming available to provide better evidence evaluation and collection. These fall roughly into two categories—tools that may be used to evaluate “dead” targets on the spot, even by relatively inexperienced investigators, and tools which permit “live” investigation, while a mission-critical machine continues to function. There are some qualms in the digital forensics community about how live forensics fits into the traditional investigative model, where exact copies of evidence (typically, hard drives) are captured and then investigated. Live machines are a moving target and there is no single “image” that defines that state of the machine. This will require some

adjustments to the investigative model, as will many of the advances on the horizon for digital forensics.

References

- Adelstein, F. (2003). MFP: The mobile forensic platform. *International Journal of Digital Evidence*, 2(1).
- Chandramouli, R., Kharrazzi, M., & Memon, N. (2004). *Image steganography and steganalysis: Concepts and practice*. Lecture notes in computer science. Springer-Verlag, Vol. 2939.
- Chandramouli, R. & Memon, N. (2003). On sequential watermark detection. *IEEE Transactions on Signal Processing, Special Issue on Signal Processing for Data Hiding in Digital Media and Secure Content Delivery*, 51(4).
- Chen, Y., Roussev, V., Richard, G. III, & Gao, Y. (2005). Content-based image retrieval for digital forensics. In Proceedings of the *First International Conference on Digital Forensics (IFIP 2005)*.
- de Vel, O., Anderson, A., Corney, M., & Mohay, G. (2001). Mining email content for author identification forensics. *SIGMOD Record*, 30(4).
- Farid, H. & Lyu, S. (2003). Higher-order wavelet statistics and their application to digital forensics. *IEEE Workshop on Statistical Analysis in Computer Vision*.
- Gao, Y., Richard, G. III, & Roussev, V. (2004). Bluepipe: An architecture for on-the-spot digital forensics. *International Journal of Digital Evidence (IJDE)*, 3(1).
- Harrison, W. (2002). A lessons learned repository for computer forensics. *International Journal of Digital Evidence (IJDE)*, 1(3).
- Mandelecha, S. (2005). *A prototype digital forensics repository*. M.S. thesis, Department of Computer Science, University of New Orleans.
- Novak, J., Raghavan P., & Tomkins, A. (2004). Anti-aliasing on the Web. In Proceedings of the *13th International Conference on the World Wide Web*.
- Patterson, D. (2004). Latency lags bandwidth. *Communications of the ACM*, 47(10).
- Roussev, V. & Richard, G. G. III. (2004). Breaking the performance wall: The case for distributed digital forensics. In Proceedings of the *2004 Digital Forensics Research Workshop (DFRWS 2004)*.
- Shanmugasundaram, K. (2003). Automated reassembly of fragmented images. In Proceedings of the *IEEE International Conference on Acoustics, Speech, and Signal Processing*.