Digital Investigation 14 (2015) S34–S44

Contents lists available at ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

DFRWS 2015 USA

Rapid forensic imaging of large disks with sifting collectors

Jonathan Grier [a, *], Golden G. Richard III [b]

[a] Grier Forensics, USA
[b] Department of Computer Science, University of New Orleans, New Orleans, LA 70148, USA

Keywords:
Digital forensics
Partial imaging
Sifting collectors
Triage
Forensic acquisition

A B S T R A C T

We present a new approach to digital forensic evidence acquisition and disk imaging called *sifting collectors* that images only those regions of a disk with expected forensic value. Sifting collectors produce a sector-by-sector, bit-identical AFF v3 image of selected disk regions that can be mounted and is fully compatible with existing forensic tools and methods. In our test cases, they have achieved an acceleration of >3× while collecting >95% of the evidence, and in some cases we have observed acceleration of up to 13×. Sifting collectors challenge many conventional notions about forensic acquisition and may help tame the volume challenge by enabling examiners to rapidly acquire and easily store large disks without sacrificing the many benefits of imaging.

The need for a new approach to forensic acquisition

The continuously increasing size and number of disk media has made digital forensics slow, cumbersome and expensive. This problem—known as the "volume challenge"—has long been identified as perhaps the greatest threat to digital forensics. Evidence has become slow and costly to acquire, store and analyze, with disks routinely taking over 10 h to image, and delays and backlogs commonplace (Richard and Roussev, 2006b; Roussev and Richard, 2004; Garfinkel, 2010; NIJ, 2014).

One innovative approach to solving this problem that has attracted considerable attention is to dispense with the full imaging process, and merge evidence acquisition with analysis, either via *live forensics* (Adelstein, 2006; Carrier, 2006) or *triage tools* (Shaw and Browne, 2013; Roussev et al., 2013; Overill et al., 2013; Marturana and Tacconi, 2013; Bogen et al., 2013). Both live forensics and triage tools involve dynamic processes for previewing the system state and available evidence, but these approaches may forgo acquisition and preservation and instead record the results of the analysis and not its sources. When on-the-spot analysis replaces acquisition, there are no longer any means to dig deeper, to look for missing clues or to pursue a contrary line of investigation beyond what was accomplished during triage. Because of these problems, imaging, despite its expense, remains the gold standard for many investigations.

Is it possible to tame the volume challenge without sacrificing the benefits of imaging? We present a new approach to imaging, called *sifting collectors*, in which disk regions with forensic value are fully duplicated to yield a sector-by-sector, bit-for-bit exact image (Section Imaging selected regions of disks), whereas disk regions that are deemed irrelevant are bypassed entirely (Section Rapidly identifying relevant regions) using either sifting profiles or investigation-driven methods (Section Alternatives to Profiles). Sifting collectors produce an Advanced Forensics Format (AFF) v3 image that can be mounted and that is fully compatible with existing forensic tools and methods (Section Storing partial images in AFF v3).

* Corresponding author.
E-mail addresses: jdgrier@grierforensics.com (J. Grier), golden@cs.uno.edu (G.G. Richard).

http://dx.doi.org/10.1016/j.diin.2015.05.006

Sifting collectors have obtained accelerations of >3× while collecting >95% of evidence and 13× while collecting >50% of evidence in controlled, quantitative laboratory trials and in real-world investigations (Section Results).

Sifting collectors as a new approach

In general, only a small portion of the data on a disk has any relevance or impact on forensic analysis. The vast majority of sectors and files contain data irrelevant to most investigations; in fact, many sectors are either blank or contain data that is found verbatim on numerous other systems (e.g., operating system and application components). Fig. 1 depicts various categories of data present on a typical disk. For some investigations, executable files may be of interest. For others, browser artifacts are of primary interest. Blank space is virtually never of use. The rest of the data, beyond what is deemed relevant to a case, and which constitutes the vast majority of the collection, could actually be replaced by random noise without affecting the forensic analysis.

Is it possible, from the outset, to identify and collect only the relevant regions of the disk? Doing so would maintain imaging's values of *preservation, verifiability, device duplication, device semantics*, and *potential for further analysis*, at a much lower cost in time. It would accelerate the collection process several times over, lower storage requirements, and accelerate analysis. Realizing this goal in an automated imaging tool, called a *sifting collector*, is our central result. This tool has two requirements:

1. It must rapidly and automatically identify the forensically relevant regions of a disk, based on an investigator's specifications or actions. More formally, it must identify the regions of the disk containing evidence, or a superset of these regions, in considerably less time than that required by conventional imaging.
2. It must image only these selected regions in a manner that is fully functional and verifiable, that preserves low-level device semantics and that is suitable for full forensic analysis. Additionally, the resultant image must be compatible with existing forensic tools and methods.

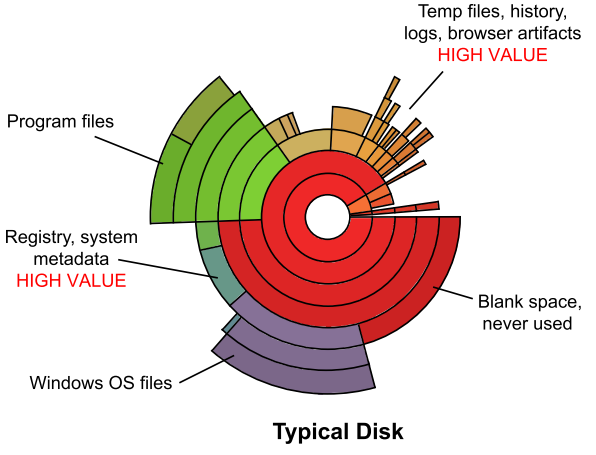

Fig. 1. Schematic illustration of typical disk. Different regions of the disk vary greatly in their forensic value.

A critical goal of our research is to *duplicate* the evidence (i.e., the physical media) and not merely record our conclusions about it. We argue that the output of triage tools, even when it includes file copies, should not be considered as a *duplicate of the evidence*, because files and their contents are not inherent properties of a disk, but rather interpretations of the disk based on filesystem software and conventions. Indeed, edge cases exist where different operating systems or tools will disagree about what the contents of a file actually are, slack space may be lost, etc. Triage tools may make these types of decisions at run time and store only their conclusions and not their sources, lacking transparency and verifiability. Note also that courts have, in some cases, deemed file by file acquisition unacceptable when drive imaging is possible (Gates Rubber, 1996).

In contrast to our goal of transparent, verifiable duplication, we do *not* maintain that collecting all the available data on a storage device is a requirement. Regardless of whether or not it is *desirable* to collect all available data, increasing storage capacities are rapidly making traditional imaging infeasible and if trends in storage capacity continue, it will ultimately be *impossible*, within reasonable resource constraints, to copy all the data. Thus, we assert that giving investigators explicit control of how much to collect is both a necessity and a virtue; such control allows investigators to analyze more devices relevant to a given case and investigate more cases with available resources. Investigating how other forensic disciplines determine which evidence is worthy of examination would be a worthwhile topic of further study; to the best of our knowledge, no other discipline routinely examines "all" possible sources without discretion. Note that we recognize different cases call for different approaches, and therefore do not argue that our goal of device level duplication of only relevant evidence is universally superior; we do argue, however, that it is a valuable and sorely needed capability.

In Section Imaging selected regions of disks, we discuss Requirement 2 (imaging only selected regions), and in Sections Rapidly identifying relevant regions and Alternatives to Profiles, we discuss Requirement 1 (rapidly identifying relevant regions).

Imaging selected regions of disks

Beginning with the first sector, we divide a disk into contiguous regions of a fixed number of sectors called *grains*; each grain is either imaged and duplicated in its entirety or completely bypassed. We image a grain if either of the following is true:

1. The grain contains any data associated with a *forensically relevant* file or the grain contains at least one forensically relevant disk sector (determining forensic relevance is discussed in Sections Rapidly identifying relevant regions and Alternatives to Profiles).

2. The grain contains any volume or filesystem metadata (such as partition tables, file tables, journals or directory structures). We collect such metadata universally because it is required to locate and read the filesystem's data and because it is typically of high forensic value. Furthermore, its relatively small size makes collection inexpensive.

The resulting collection is thus a full, bit-for-bit duplication of selected regions of the disk, preserving low-level device semantics and critical qualities of imaging. Unlike most triage tools, we preserve only raw, device-level data that are directly verifiable against the original media; we do not record or preserve any forensic analysis or even filesystem-level abstractions such as file contents. Thus, we maintain the valuable properties of imaging, while avoiding the time-consuming duplication of blank or irrelevant regions of the disk.

Using *sifting profiles* (described in Section Rapidly identifying relevant regions), which describe files of interest, we are able to rapidly identify the regions meeting these criteria by leveraging a universal property of filesystems: By necessity, all filesystems must contain data structures that readily identify their files, their properties and their associated sectors; these structures are required for rapid file access during normal operations. We use these very structures to locate metadata and to rapidly enumerate all files on a disk, to examine their properties and determine their forensic interest and, when warranted, to identify the sectors associated with them. Our prototype implementation targets NTFS and uses the Master File Table (MFT) as its primary source; the approach, however, should work equally for other filesystems. In many cases, even when files have been deleted, their associated metadata structures remain, which allows our implementation to function normally (Carrier, 2005). When the metadata has been overwritten, files can sometimes still be recovered using *file carving* (Richard and Roussev, 2005); although our prototype was not intended to support file carving, carving nonetheless succeeded in over 75% of the cases we tested (see Section Results). For cases where profiles are not appropriate, Section Alternatives to Profiles describes another investigation-directed method for rapidly identifying sectors of interest.

Because the resulting image is a device-level duplicate and includes all volume and filesystem metadata, it can be mounted using standard tools and examined using standard forensic procedures. We discuss below how we indicate the absence of the bypassed regions in the image and how we handle read requests to those regions (Section Storing partial images in AFF v3), and describe how we use the collected image with FTK, the Sleuthkit, *log2timeline* and other standard forensic tools (Section Results).

Our approach is *not* suitable for physically damaged media, or damaged, tampered with or unknown filesystems. In the future, we intend to incorporate a media integrity check into the collector, which reverts to conventional imaging in these cases. Likewise, our approach is not suitable for steganographic filesystems, such as Rubberhose (Rubberhose, 2015) or StegFS (StegFS, 2015), or other cases where files have been hidden beyond the reach of standard operating systems; to our knowledge, such systems, however, have not been widely adopted in the wild.

Quantifying a region's forensic relevance

Before exploring how to rapidly identify relevant sectors, we first introduce an objective, observable measure of relevance. Consider the following:

Definition 1. A region is *forensically relevant* if and only if the conclusions of an associated investigation are substantially altered if the region's contents are replaced with random values.

Relevance is therefore not a property of a region alone, but a *region in the context of a forensic investigation*. Specifically, it depends on the *purposes* of investigation (Pogue, 2011). Despite its dependence on context, relevance can indeed be quantified, using the following abstract procedure:

1. Perform a forensic investigation in a fully reproducible manner. That is, the procedure must be an *effective procedure*, using Turing's classic definition (Turing, 1936), i.e., a fully defined or automated procedure which results in a quantifiable, definite conclusion.
2. Randomize the value of a particular region.
3. Repeat the investigation. The region is relevant, in the context of this investigation, if and only if, the conclusions of the investigation differ.

We now have an objective, observable definition of relevance, as a function of both the data in the region and the investigative procedure employed.

Definition 2. A region which is never read in the course of a forensic analysis procedure is called *manifestly irrelevant.*

Note that some analysis procedures, such as a full disk search for keywords, leave *no* region manifestly irrelevant.

The concept of relevance can be extended to a region which has not yet been read, but is known to have certain properties. We call this concept *expected relevance*, and quantify it as a probability measure (between 0 and 1), defined as the a priori probability that a region known to have certain properties is relevant to a particular investigation. Intuitively, this can be understood as the probability that a region with such properties on a randomly selected disk will be relevant (using the frequentist probability interpretation), or our degree of belief that a particular region known only to have such properties is relevant (using the Bayesian probability interpretation) (Hájek, 2003). Expected relevance is likewise a function of both the known properties of that region and the investigative procedure employed.

Given sufficient cases, disks, and reproducible forensic procedures, it is possible to measure expected relevance over numerous types of regions. In general, we conjecture that:

Proposition 3a. Regions which have *never* been allocated have very low expected relevance (close to 0).

Proposition 3b. Regions containing volume or filesystem metadata have very high expected relevance (close to 1).

Proposition 3c. Regions known to contain data belonging to known artifacts (e.g., a browser cache), or to known locations of artifacts (e.g., a log directory) have high expected relevance.

Proposition 3d. The expected relevance of a region is highly correlated with the file types and file names whose data is contained in the region.

Proposition 3e. In general, the expected relevance of a region is highly correlated with properties that can be ascertained from filesystem metadata, such as file names, types, and locations of data stored within the region; MAC times associated with the region; and allocation status and history of the region.

Our approach is based on these propositions and the results of our initial case studies (described in Section Results), support them. In future work, they will be measured over a larger sample space. We note that filesystem metadata makes it easy to identify currently allocated sectors, but does not always allow differentiating previously allocated sectors from never allocated sectors. However, given that sector allocation is often sequential, it is possible to infer from the metadata the highest sector that was ever allocated (since the filesystem structures were created), within a reasonable margin of error. This inference can also be tested in real-time with sampling. This allows for a simple, low risk approach to eliminating never allocated sectors from consideration.

Rapidly identifying relevant regions

As we've argued previously, only a small part of a typical collection is ultimately relevant. How do we automatically identify these regions of the disk at the time of imaging? For that matter, how do forensic examiners ultimately identify such regions manually? Can we incorporate the procedures that such forensic examiners use into the imaging process itself?

A common process underlies most forensic examinations: forensic examiners or automated tools find relevant data; the examiners (or tools) parse and analyze these relevant data; and the examiners (or tools) learn the location of additional forensic data from these data, and the process then repeats recursively. For example, in Lee's examination of a compromised IIS Web server (Lee, 2012), he first uses the Sleuthkit to parse the partition table and find the filesystem's initial sector. With that information, he locates and parses the MFT in the NTFS partition and uses the MFT to locate the system logs, which he further analyzes using the log2timeline tool (Log2timeline, 2015). Formalizing this process, we can view a disk as a graph, with sectors as vertices and data references as edges. The essential concept is that we can find all necessary data by walking a spanning tree (see Fig. 2). Binary reverse-

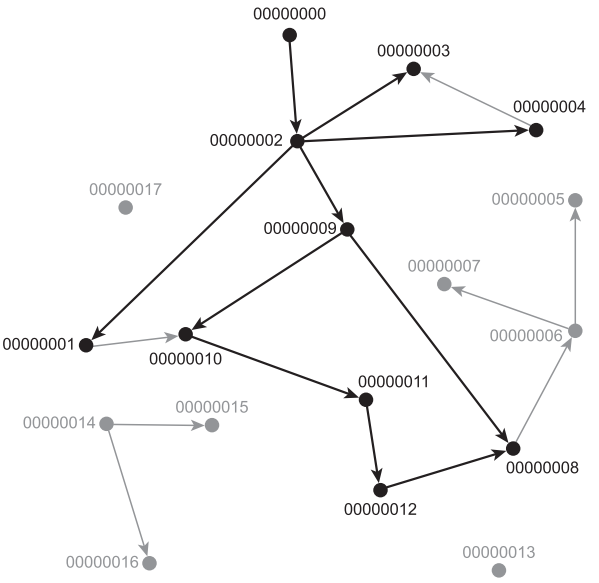

Fig. 2. A disk can be abstractly viewed as a graph, with sectors as vertices and data references as edges. Forensically relevant sectors and references (shown in bold) form a spanning tree. By walking this spanning tree, we can image all necessary data without reading the entire disk.

engineering tools such as IDA Pro use a similar algorithm to identify executable code (Bachaalany, 2011; Pyew, 2015); to the best of our knowledge, no one has yet explicitly noted the algorithm's relevance to disk forensics. Our goal is to capture this process in an automated method.

To automate this scheme as part of the imaging process itself, we introduce the concept of a *focusing procedure*. A focusing procedure is any means by which an examiner identifies certain data as relevant and chooses to ignore other data as irrelevant. To incorporate a focusing procedure into an imaging tool, the procedure must have two properties:

1. It must not require spontaneous ingenuity or judgment but must be definable in advance.
2. It must not require exhaustive reads of the entire disk but must use some form of pointers, index, metadata, or other mechanism to directly identify the location of the applicable data.

We call a focusing procedure with both these properties *compressible*; see Table 1 for some common examples. Clearly, many focusing procedures, such as examining known artifacts or files with timestamps matching a window, are fully compressible. However, other focusing procedures are not fully compressible: for example, keyword-based disk searches lack the second property because such searches typically require reading the entire disk.

Compressible focusing procedures can readily be incorporated into the imaging process itself, collecting only those sectors identified as relevant. Using this exclusively would entail forgoing all non-compressible focusing procedures, resulting in functionality that resembles a triage tool in that it delivers the most readily obtainable evidence,

Table 1
Common focusing procedures. Procedures with YES values for both properties are compressible and can be performed at acquisition time as part of the imaging process itself.

Focusing procedure	Example	Can it be defined in advance?	Can it be performed with minimal disk reads?
Known artifact	Examine known artifacts, such as Firefox's history, stored in known folders	Y	Y
MAC timestamps	Examine only files created within a certain timeframe	Y	Y
Anomalous file name or location	Examine files in the WINDOWS directory that are anomalous or misspelled	Y	Y
Autoruns	Examine files that run automatically on boot	Y	Y
Hash-based file search	Examine files with hashes corresponding to known malware or contraband	Y	N
Keyword-based sector search	Search disk for keywords of interest	N	N

but still maintains some of the benefits of imaging (e.g., device duplication and verifiability). To collect evidence more comprehensively than attainable this way, we also use an additional approach, called *supersetting approximations*, described below.

Supersetting approximations can best be illustrated with a concrete example. Keyword-based sector searches are not compressible because any sector may possibly contain a keyword. Nonetheless, the probability distribution of containing a keyword is by no means uniform over all sectors. Sectors that are associated with text and document files, for example, have a high probability of containing a keyword; sectors associated with executables, for example, or sectors that have never been allocated, have a much lower probability of containing a keyword. Although a keyword search is not compressible, we can readily estimate the probability of a sector matching and collect an *approximating superset* of sectors with a sufficiently high probability of matching. For example, an exhaustive keyword search of an entire disk might yield 100 sectors, whereas a probabilistically defined approximating superset of such disk might contain 100,000 sectors, including 98 of the true 100 sectors that can be found via the full keyword search.

As explained above in Section Quantifying a region's forensic relevance, relevance is defined in the context of a particular mode of investigation. Consequently, profiles must be selected (or authored) to match the needs of the particular case. Initially, we intended to develop a complex, procedural language to define supersetting approximations. However, we discovered that such complexity was not warranted, since expected relevance is sufficiently correlated with simple, readily determined properties, often expressible as simple regular expressions. Although it may be possible to use procedural definitions to tighten these supersets, we prefer the simplicity gained by using simple expressions.

Our implementation, written in C++, allows an examiner to select a pre-existing (or write a fresh) *sifting profile* that defines file relevance based on these readily determined properties, such as file name, location and timestamps. *MIME type*, such as *text document* or *audio*, is a key driver of expected relevance; we use file name, extension, and location as proxies. For many investigations, only files created and edited by users, or edited during certain time periods, are relevant (Pal and Memon, 2009). We determine these by observing MAC timestamps, as described in (Farmer and Venema, 2005; Agrawal et al., 2007). Our prototype targets NTFS and can typically determine the sectors associated with matching files by simply scanning the MFT. We have currently implemented a small library of profiles that are targeted to collect different types of

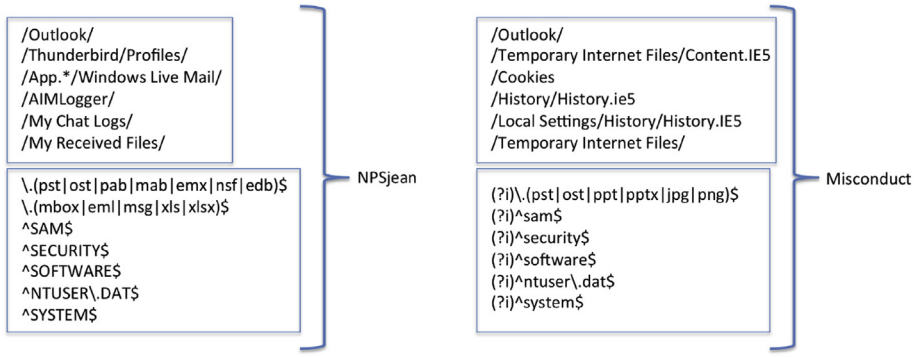

Fig. 3. Some representative profiles. The NPSjean profile was used to solve the popular open source case and targets various email and chat artifacts as well as registry files and spreadsheets. The Misconduct profile targets browsing history, Outlook email, and various graphics file formats. The components of a profile define relevant pathnames and filenames.

evidence for common classes of investigations and are actively working to expand this library. Representative profiles are depicted in Fig. 3, to illustrate their basic structure, which currently consists of descriptions of relevant pathnames and filenames, based on regular expressions, and boolean expressions concerning MAC timestamps. In writing the profiles, we were guided by both existing forensic practice and size distributions of common file types. In particular, the irregular bimodal distribution of file sizes (Agrawal et al., 2007) indicates that rejecting even a small number of file types can result in large savings of space; thus, we are liberal in collecting files that are typically small, and frugal in collecting files that are often large.

Our testing shows that by using supersetting approximations, we are able to recover between 54% and 100% of relevant evidence for a case in a fraction of the time (Section Results). Supersetting approximations work for two reasons: First, because exhaustive searching is impractical for daily operations, systems typically include rapidly accessible metadata to locate data of interest; as long as we can find and parse that metadata, we can identify which sectors are likely to have the data of interest without having to read the sectors themselves. Second, due to the locality principle (Denning, 2005), forensically relevant sectors are frequently adjacent to other forensically relevant sectors; thus, if we can find metadata that indicates the expected relevance of even one sector in the region, we can collect the entire region. Therefore, we can tolerate the absence of some metadata without losing evidence. Due to the locality principle, our implementation, though not intended to support file carving, allowed carving operations to succeed over 75% of the time (see Section Results).

One important weakness of sifting collectors is their increased vulnerability to steganography and anti-forensics (Foster and Liu, 2005). Anti-forensics (for example, disguising a suspicious file to appear innocuous), though effective even against conventional imaging, is especially potent when applied against sifting collectors. When used against conventional imaging, anti-forensics can cause an analysis failure, whereas with sifting collectors, it causes the evidence to be completely omitted from collection. If an analyst using conventional imaging suspects anti-forensics might have been deployed, they can develop and apply countermeasures at any point in the process (Garfinkel, 2007), whereas using sifting collectors, these countermeasures must be applied at, and incorporated into, the collection process itself. Moreover, when supersetting approximations are used, anti-forensics is made fundamentally easier, since only the metadata, and not the data itself, needs to be disguised. For example, under conventional imaging, a file disguised to appear innocuous may in some cases still be found through exhaustive hashing, which is incompatible with supersetting approximations. Finally, if sifting collectors were to become widely used, it's possible that steganography and anti-forensics specifically designed to hide from sifting collectors would be developed. Although we intend in future work to incorporate anti-forensics countermeasures into sifting collectors, for these reasons we feel that sifting collectors' exposure to anti-forensics will remain higher than conventional imaging, yet lower than triage tools.

Alternatives to profiles

In many cases, profiles (discussed in the previous section) provide a convenient mechanism for identifying relevant regions of a storage device. In cases where triage is preferred (or required), we offer a different method, which essentially endows triage efforts with *reproducibility*. This involves introduction of a "human-in-the-loop" partial imager (HILPI). HILPI is inserted into the access path of the storage device to be examined and a virtual disk is exposed to the examiner which shadows the target storage device, as illustrated in Fig. 4. Using a copy-on-read (COR) strategy, all sectors that are read from the virtual disk during the examination are automatically copied into an AFFv3 format sifted image. An investigator using HILPI is free to explore the disk in any way they choose. However, unlike typical live forensics scenarios, a complete chain of evidence of the state of the disk, at the device level, is preserved and imaged, in a format which allows reexamination and full reproducibility. For example, imagine an examiner uses a tool to locate all files on a disk modified during a certain time window. The tool reads and queries the filesystem metadata stored on the disk and as these reads occur, HILPI automatically creates a partial image of these sectors, allowing subsequent verification of this analysis. The examiner then selects one file from the list, examines it,

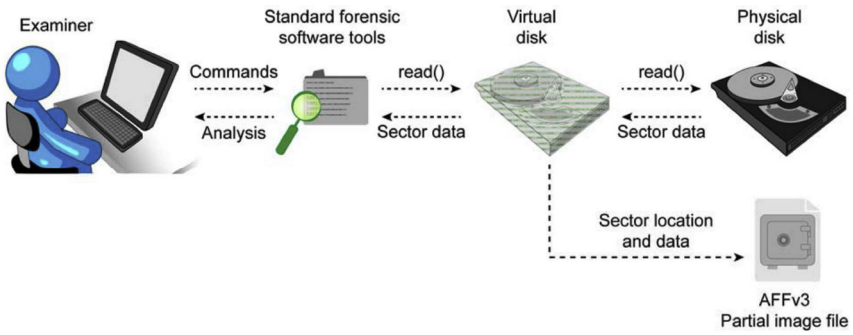

Fig. 4. Our "human in the loop" imager (HILPI), which uses a copy-on-read (COR) strategy to duplicate only regions of a disk that are actually accessed during a forensic investigation to an AFFv3 sifted image. HILPI can either supplement or replace the use of traditional profiles in our partial imaging scheme.

and finds no evidence of worth. Again, all sectors of the disk used to read the file (including both the files data sectors and relevant metadata sectors) are automatically imaged by HILPI, allowing an opposing examiner to verify or dispute this judgment. Finally, the examiner opens a second file and draws conclusions from it. Without manual action, HILPI ensures that all sectors required to reconstruct this file are preserved. Any sector that is examined, explicitly or implicitly, or in any way influences the examiner's live investigation, is preserved in the image. And, while the sectors that the examiner never reads are likewise omitted, whatever process the examiner uses to determine that these sectors are not relevant is fully reproducible, allowing full defense or challenge to the process, without the need to rely on the examiners' faith, credibility, or memory. In this manner, HILPI brings the verifiability and reproducibility long enjoyed by forensic imaging to live forensics, while preserving live forensics' speed and power. After the examination is completed, the AFFv3 format sifted image is available to permit the same tools (and any other tools which access the same data that was collected) to be used again to demonstrate the results obtained during triage (see Fig. 5).

Storing partial images in AFF v3

We store the resulting image in the Advanced Forensics Format (AFF) v3 format (Garfinkel et al., 2006). v3 is the most commonly used version of AFF, and was not designed or intended for partial images. Nonetheless, we are able to use it for such without modification, as explained below. By repurposing an existing, widely supported format, we gain compatibility with a wide array of existing forensic tools and procedures, avoiding the barriers to adoption that come with introduction of a new format.

AFF consists entirely of uniquely named *segments* that store either the raw device data or metadata about the image. Segments are retrieved by name via a master dictionary and can thus be stored in any order. Device data are grouped into contiguous regions called *pages*, and each page is stored in a single, uniquely named segment (e.g., segment `page30 may store bytes 251658240–260046847`). Within a segment, data are stored and retrieved by position, providing good performance.

Although not designed for partial images, AFF's elegant design can fortuitously be used to support such images: as long as the sifting collector's grain size equals the AFF format's page size (or an even multiple of it), all AFF pages will be either imaged in their entirety or completely absent. Therefore, we can simply omit uncollected pages from the image file.

The AFF implementation used in tested versions of AccessData's FTK functioned as-is with our enhanced AFF partial images. The AFFLIB v3 implementation required only a simple patch. For forensic tools that do not support AFF, we use existing loopback adapter interfaces (affuse on Linux and FTK's built-in adapter on Windows), and these tools allowed all our tested forensics tools to operate flawlessly.

We thus propose a single, fully backwards compatible addition to the AFFv3 specification:

A data page MAY be absent from an AFFv3 file. This absence occurs when the page was not collected (i.e., a partial image). Implementations MUST therefore appropriately handle read requests for data located on a missing page. Implementations SHOULD be configurable to handle this situation through any of the following responses:

1. *Returning an error indicating the data are missing*
2. *Returning well known dummy data*
3. *Returning binary NULLs*
4. *Indicating a bad sector.*

Finally, as an alternative to AFF, we support writing RAW files using sparse null data for bypassed regions. As discussed in Section Related work, however, sparse files can lose storage savings when copied, and therefore we don't encourage using this format.

Results

To test the accuracy, time savings and compatibility of our approach with existing forensics tools and methods and to measure the quantity and quality of evidence collected, we performed a battery of controlled tests on both sifted collections (collected by our C++ prototype implementation) and on standard images as a control. The tests included both laboratory testing and recreating two full digital investigations. We performed our tests using common, unmodified forensic tools and compared the investigative results with the results from ground truth and detailed investigations performed on full disk images, as appropriate. We discuss our results in this section.

Disk	Profile	Acceleration	Tool	Accuracy	Comprehensiveness
NIST CFReDS Hacking Case (4.6 GB)	Registry	4.5x	log2timeline	100%	100%
	Registry	4.5x	Regripper	100%	100%
	IEHistory	5.0x	Pasco	100%	100%
	Email	3.7x	Bulk_extractor	100%	95%
	Registry	4.5x	Mactime	100%	100%
NPS DOMEXUSERS (40 GB)	Registry	13x	log2timeline	100%	54%
	Registry	13x	Regripper	100%	100%
	IEHistory	13x	Pasco	100%	100%
	Email	11x	Bulk_extractor	100%	57%
	Registry	13x	Mactime	100%	100%

Fig. 5. Quantitative test results for the CFReDS and DOMEXUSERS cases.

We computed the acceleration factor as the total number of disk sectors read during conventional imaging divided by the total number of disk sectors read during sifting collection. Disk reads are the bottleneck in high-speed imaging and our implementation is designed to exploit disk characteristics and minimize search time by performing reads in disk order in 8 MB contiguous batches; thus, we believe that this calculation is appropriate. For future research, we intend to build a multithreaded collector with optimized code and compression routines and benchmark the acceleration in clock time.

Accuracy testing

In evaluating the accuracy of a sifting collector, the critical issues are whether the partial image acquired by the sifting collector is accurate and precisely mirrors data on the original disk, whether the partial image contains all data targeted by the selected profile, and whether the AFF format of the partial image is robust and complete. For our accuracy tests, we ran standard forensic tools against both sifted and unsifted versions of a number of publicly available images and performed tests at the volume, filesystem, file and sector levels. At the volume and filesystem levels, we queried metadata using mmls and fsstat from the Sleuthkit and verified that the results matched exactly. The only discrepancies noted at the volume level resulted from the sifting collector padding volume lengths with null bytes to a multiple of 8 MB, which has since been corrected. At the individual file level, we tested many files and directories using istat and icat. All the tests were successful except for tests involving NTFS compressed files, which are not currently supported. At the sector level, a large number of random sectors were selected and evaluated on a bit-by-bit basis using dd and affcat. All tests at the sector level indicated that each sector was collected accurately.

Quantitative testing

We further measured the accuracy, acceleration, and comprehensiveness by using existing forensic tools on two publicly available source disks, NIST CFReDS (CFReDS, 2015) and NPS nps-2009-domexusers (NPS-DOMEXUSERS, 2015), comparing the results of bulk_ extractor (Bulk_extractor, 2015), log2timeline (Log2timeline, 2015), pasco (Pasco, 2015), regripper (Regripper, 2015), and the complete set of the Sleuthkit (TSK, 2015) command line tools. Because they collect large numbers of data (e.g., finding every email address on the disk), these tools can objectively quantify the portion of evidence collected; thus, we determined that bulk_extractor was able to find 26,390 emails on the conventional image and 25,111 emails on the sifted image. On the CFReDS disk, we achieved an acceleration between 3.7x-5.0× and collected between 95 and 100% of the evidence with 100% accuracy. On the DOMEX USERS disk, we achieved an acceleration between 11x-13×, collecting between 54% and 100% of the evidence at 100% accuracy (see Fig. 5). Preliminary investigation indicates that the missing data (in the case of 54% collection) were located in the System Restore Points, which the collector was not configured to acquire.

Qualitative testing

Because a key requirement is compatibility with existing forensic tools, we used popular GUI tools with the images, such as FTK v3.2, FTK Imager Lite and osTriage (using FTK's adapter). Because sifted images use the standard AFF format, the images were processed exactly as expected by all three tools; disk regions not present in the sifted image were represented by zero-filled content, and all the files targeted by the selected profile were correctly processed.

Investigation # 1: A synthetic case: M57-Jean

The M57-Jean case (M57-Jean, 2015) involves a startup company where an Excel spreadsheet with confidential information, originating from their CFO's computer, was discovered posted in the technical support forum of a competitor. The investigation must determine when the spreadsheet was created, how it got to the competitor's website and who (besides the CFO) might be involved in the data exfiltration. The evidence consists of a copy of the spreadsheet and a single 10 GB NTFS disk. Because the investigation targets Excel spreadsheets, employee communications, and other potential avenues for data exfiltration, a sifting collector profile (named NPSJEAN) was developed that targets Outlook, Thunderbird, and Windows Live email; common chat clients, such as AIM; and Windows registry files, which are important in analyzing USB devices previously connected to a computer running Windows. A sifting collector was used with this profile and imaged the disk with an acceleration factor of 3.2×. An investigation was then initiated using the Sleuthkit (modified only by linking with our updated version of AFFLIB), readpst (Libpst, 2015), regripper, and other tools. The contested spreadsheet and a number of relevant emails and chat logs were discovered. USB activity was inconclusive, although the previous attachment of two USB devices was noted. Examination of the emails and chat logs indicates a strong likelihood of a phishing attack against the CFO that resulted in data exfiltration; however, chat messages reveal the possibility that the CFO's account had been compromised. We wrote up our investigative conclusions in a final report, which we then compared against the publicly available solution: our conclusions and discoveries matched exactly. Sifting collectors thus yielded an acceleration factor of 3.2×, and 100% collection accuracy and comprehensiveness.

Investigation # 2: A real case: employee misconduct

We also investigated a real case from our previous private casework, involving employee violations of corporate policies against workplace viewing content that is not suitable for work (NSFW). Outlook email (including attachments), photographic images, relevant Windows registry artifacts and the use of Internet Explorer were targeted by the profile. The original investigation involved three disks (40 GB, 160 GB and 320 GB in size), and involved recovering and investigating both intact and deleted pictures, emails, documents, browser histories, and other artifacts, to discover evidence of NSFW material. It used FTK

v3.2 on Windows 7 to process and index the raw disk images, file carving to recover deleted files, a review of approximately 185,000 pictures, and review and bookmarking of discovered Outlook email.

To perform this investigation using sifting collectors, we developed a profile named MISCONDUCT, targeting photographic images, the Windows registry, web-browsing artifacts and Outlook email, and used it to collect each disk. Sifted disk images were successfully collected for the 160 GB and 320 GB drives, but failed for the 40 GB disk, which appears to have been caused by disk corruption (another forensic tool was also unable to process the disk, although FTK v3.2 processed it correctly in the original investigation). Fortunately, the original investigation (using conventional imaging) found no usable evidence on the 40 GB disk, and the disk was omitted from further consideration. Sifting collectors achieved a 2.9× acceleration factor on the 160 GB disk, and a 9.6× acceleration factor on the 320 GB disk.

We repeated the original investigation process on the sifted images, using the same forensic tools and environment as the original investigation, and compared their results. We recovered and analyzed photographs, browser history, Powerpoint presentations, Outlook emails, and registry artifacts, and performed limited file carving. The sifting collector investigation yielded 100% of the relevant evidence for the 320 GB disk, which contained 177,775 files in total. 100% collection comprehensiveness is attributed to the fact that file carving in the original case yielded no NSFW files for the 320 GB disk. For the 160 GB sifted AFF disk image, 62 JPEG files (out of 10,193 JPEGs that did not require carving) and 32 PNG files (out of 6495 that did not require carving) that were expected to be collected by the sifting collector were not included in the image. Analysis revealed that these missing files were in a particular branch of the Internet Explorer history directory; we are currently investigating the cause of this omission. Moreover, although our implementation was not designed to support file carving, we were nonetheless able to recover 17 out of the 22 NSFW carved photographs that were identified in the original investigation; we attribute this to these files being located in adjacent regions to files with intact metadata, as described in Section Rapidly identifying relevant regions. Thus, for the 160 GB disk, only 95 files that did not require carving and 5 files that did require carving, out of a total of 322,364, were missing from the sifted image, yielding >99% collection comprehensiveness.

In summary, the sifting collector yielded an acceleration factor of 2.9× and a collection comprehensiveness of over 99% on the 160 GB disk, and an acceleration factor of 9.6× and a collection comprehensiveness of 100% on the 320 GB disk.

How useful are sifting collectors to forensic examiners?

We informally interviewed forensic examiners to learn the practical value that sifting collectors may provide. Examiners reported highly divergent attitudes about both their need for imaging and the problems caused by its long duration. We were initially perplexed by this variance, until we grouped examiners into three segments (law enforcement, civil litigation and incident response), each of which displayed internal consistency (see Table 2). Sifting collectors deliver the most value for examiners with a high need for imaging and high sensitivity to its duration (as observed in law enforcement).

Table 2
Different segments of forensics community. Sifting collectors deliver the most value for users with both time pressure and the need for preservation and verifiability.

User segment	Time pressure	Need for preservation, verifiability, and admissibility
Law enforcement	High	High
Civil litigation	Low	Very high
Incident response	Very high	Low

As implemented, sifting collectors require a profile defining the types of evidence to collect. We do not expect examiners to be able to write such profiles anew for each case but instead to select from a library. This approach relies on what we call the *investigation class hypothesis*: investigations fall into a limited number of classes, each with similar evidentiary requirements. Testing this hypothesis is a key goal of future research.

Related work

For over a decade, researchers have raised concerns over the inadequacy of forensic processes to address increasing disk volume and the need for alternatives to full acquisition procedures (Richard and Roussev, 2006b, a). (Botchek, 2008) provides an excellent discussion of the performance limits of disk imaging. Some have argued that, in addition to practical concerns, conventional imaging is exposed to significant legal risks due to the inability to turn around cases in a timely manner (Kenneally and Brown, 2005). Turner's digital evidence bags, which mimic containers used in other forensic disciplines (Turner, 2006), offer one of the earliest examples of selective imaging. AFFv4 introduces explicit support for composing images out of multiple sources and regions, making it well suited for partial imaging. However, AFFv4 has not seen adoption, largely due to its complexity. Our work demonstrates that existing, simple, widely supported formats, though not designed for it, can fully support partial imaging. Stuttgen clarifies the distinction between collecting at the file, filesystem, partition and device (i.e., sector or block) levels, and presents software allowing an investigator to choose particular sectors to image (Stuttgen et al., 2013). However, this is a manual approach, in which the investigator manually chooses each individual sector or file that is desired. Moreover, because it may lack sectors with filesystem metadata, the resulting image is not necessarily mountable or analyzable by standard forensic tools.

X-Ways Forensics (X-Ways, 2015), a commercial forensic tool suite, also offers selective acquisition via a feature called *skeleton images*. Regions and data to be copied must be selected manually, similar to that of Stuttgens tools, so

that these skeleton images save storage space, but do not save time or accelerate the collection process. Unlike sifting, which first does a rapid analysis pass, determines sectors of interest, and collects them in disk order, disk sectors are collected as items are manually selected, causing additional bottlenecks. Finally, skeleton images are stored as NTFS sparse files, which means that when images are copied to different media, all storage savings will be completely lost. In contrast with manual approaches like Stuttgen's and X-Ways, sifting collectors automate *and accelerate* selective acquisition. Furthermore, by repurposing the standard AFF format to support partial images, sifting collectors produce portable images that are readily analyzable by existing tools, without modification.

Triage has become a hotly debated topic in the digital forensics community (Roussev et al., 2013; Overill et al., 2013; Marturana and Tacconi, 2013; Bogen et al., 2013), with the Journal of Digital Investigation devoting an entire issue to it (Digital Investigation, 2013). While certainly valuable, triage has several shortcomings as a replacement for acquisition, such as missing important evidence and potentially damaging relevant evidence, particularly in view of how triage is currently practiced (Shaw and Browne, 2013). In fact, some researchers argue that triage is a focus of study simply because the only other alternative appears to be complete imaging (Pollitt, 2013). We submit that sifting collectors offer another option by preserving the qualities of imaging (and the potential for deep forensic analysis in the lab) while addressing the concerns of the volume challenge and backlogs.

Future work

Testing sifting collectors in the real world is the next step. This testing requires expanding the library of profiles, improving the approach's robustness and conducting controlled tests on real-world cases. We are currently implementing a multithreaded, optimized collector and when this is complete, will produce timed performance benchmarks. In addition to profiles and HILPI, we are also investigating other means of identifying relevant disk regions, including random sampling (Garfinkel et al., 2010), file-level analysis, and integration with memory forensics (e.g., identifying files from memory). We are currently maturing our lab prototype into a production ready, robust engine, with a goal of direct integration into existing forensic products. We aim to integrate sifting collectors into the gamut of forensic technology, including disk duplicators and imagers (both hardware and software based), analysis tools (allowing post-collection reduction of image size), triage tools and live forensics (so that a sifted image, allowing full reproduction of the triage, is generated as the triage takes place) and enterprise forensics and incident response (where scarce bandwidth makes sifting acceleration particularly valuable). We invite community members interested in participating in such to contact us.

Conclusions

The volume and backlog of digital evidence continues to grow, and novel paradigms of acquisition are required. We argue that sifting collectors present such an innovation. Moreover, sifting collectors challenge several commonly held notions: Previously, the question of *what* evidence to acquire (entire devices versus selected data) was assumed to be intertwined with the question of *how* to acquire it (media duplication versus acquiring files or live analysis). Sifting collectors disrupt such coupling. Likewise, they provide new opportunities to innovate by demonstrating that novel modes of acquisition can still support device-level forensics and be fully compatible with existing forensic tools.

Acknowledgement

This work was supported in part by DARPA and by the National Institute of Justice Award 2014-IJ-CX-K001. The authors would also like to thank Dan Farmer and the anonymous reviewers. All opinions expressed are solely those of the authors.

References

Adelstein F. Live forensics: diagnosing your system without killing it first. Commun ACM 2006;49(2):63–6.
Agrawal N, Bolosky WJ, Douceur JR, Lorch JR. A five-year study of file-system metadata. ACM Trans Storage (TOS) 2007;3(3):9.
Bachaalany E. Why there is no undo option in IDA Pro. 2011. http://www.hexblog.com/?p=415.
Bogen PL, McKenzie A, Gillen R. Redeye: a digital library for forensic document triage. 2013. p. 181–90.
Botchek R. Benchmarking hard disk duplication performance in forensic applications. 2008. http://www.tableau.com/pdf/en/Tableau_Forensic_Disk_Perf.pdf.
Bulk_extractor, 2015. Bulk_extractor tool. https://github.com/simsong/bulk_extractor.
Carrier B. File system forensic analysis. Addison-Wesley; 2005. http://books.google.com/books?id=I4gpAQAAMAAJ.
Carrier BD. Risks of live digital forensic analysis. Commun ACM 2006;49(2):56–61.
CFReDS. NIST CFReDS disk images. 2015. http://www.cfreds.nist.gov/dfr-test-images.html.
Denning PJ. The locality principle. Commun. ACM Jul. 2005;48(7):19–24. http://doi.acm.org/10.1145/1070838.1070856.
Farmer D, Venema W. Forensic discovery. Addison-Wesley Upper Saddle River; 2005.
Foster JC, Liu V. Catch me, if you can. Blackhat; 2005.
Garfinkel S. Anti-forensics: techniques, detection and countermeasures. In: In: the 2nd international conference on i-warfare and security (ICIW); 2007. p. 77–84.
Garfinkel SL. Digital forensics research: the next 10 years. Digital Investigation 2010;7:S64–73.
Garfinkel S, Malan D, Dubec K-A, Stevens C, Pham C. Advanced forensic format: an open extensible format for disk imaging. In: Advances in digital forensics II. Springer; 2006. p. 13–27.
Garfinkel S, Nelson A, White D, Roussev V. Using purpose-built functions and block hashes to enable small block and sub-file forensics. Digital Investigation 2010;7:S13–23.
Hájek A. Interpretations of probability. In: the Stanford encyclopedia of philosophy. Citeseer; 2003.
Digital Investigation. Triage in digital forensics. http://www.sciencedirect.com/science/journal/17422876/10/2; 2013.
Kenneally EE, Brown CL. Risk sensitive digital evidence collection. Digital Investigation 2005;2(2):101–19.
Lee R. Digital forensic sifting: targeted timeline creation and analysis using log2timeline. 2012. http://computer-forensics.sans.org/blog/2012/01/20/.
Libpst, 2015. libpst tools. http://www.five-ten-sg.com/libpst.
Log2timeline, 2015. log2timeline timelining tool. http://log2timeline.net.
M57-Jean. M57-Jean forensic scenario. 2015. http://digitalcorpora.org/corpora/scenarios/m57-jean.
Marturana F, Tacconi S. A machine learning-based triage methodology for automated categorization of digital media. Digital Investigation 2013;10(2):193–204.

NIJ. New approaches to digital evidence processing and storage. U. S Department of Justice; 2014.
NPS-DOMEXUSERS. NPS DOMEXUSERS disk image. 2015. http://digitalcorpora.org/corp/nps/drives/nps-2009-domexusers/.
Overill RE, Silomon JA, Roscoe KA. Triage template pipelines in digital forensic investigations. Digital Investigation 2013;10(2):168–74.
Pal A, Memon N. The evolution of file carving. Signal processing Magazine, IEEE 2009;26(2):59–71.
Pasco, 2015. pasco tool. http://www.mcafee.com/us/downloads/free-tools/pasco.aspx.
Pogue C. Sniper forensics, part 1: a brief history lesson. 2011. http://blog.spiderlabs.com/2011/01/spiderlabs-blog-post-sniper-forensics-part-1.html.
Pollitt MM. Triage: a practical solution or admission of failure. Digit Investig 2013;10(2):87–8.
Pyew, 2015. pyew: a python tool for static malware analysis. http://code.google.com/p/pyew/wiki/CodeAnalysis.
Regripper, 2015. regripper registry analysis tool. http://regripper.wordpress.com.
Richard GG, Roussev V. Scalpel: a frugal, high performance file carver. In: Digitalal Forensics Research Conference (DFRWS 2005); 2005. p. 71–7.
Richard GG, Roussev V. Digital forensics tools: the next generation. Digital Crime and Forensic Science in Cyberspace. Idea Group Publishing; 2006a. p. 75–90.
Richard GG, Roussev V. Next-generation digital forensics. Commun ACM 2006b;49(2).
Roussev V, Richard GG. Breaking the performance wall: the case for distributed digital forensics. In: Proceedings of the 2004 Digital Forensics Research Workshop; 2004.
Roussev V, Quates C, Martell R. Real-time digital forensics and triage. Digital Investigation 2013;10(2):158–67.
Rubber Gates. United States district court for the district of Colorado. Gates Rubber Co. v. Bando Chemical Indus., Ltd; 1996.
Rubberhose. Rubberhose steganographic filesystem. 2015. http://en.wikipedia.org/wiki/Rubberhose_(file_system).
Shaw A, Browne A. A practical and robust approach to coping with large volumes of data submitted for digital forensic examination. Digital Investigation 2013;10(2):116–28.
StegFS. StegFS filesystem. 2015. http://en.wikipedia.org/wiki/StegFS.
Stuttgen J, Dewald A, Freiling FC. Selective imaging revisited. In: Seventh international conference on it security incident management and it forensics; 2013. p. 45–58.
TSK. The sleuthkit (TSK). 2015. http://www.sleuthkit.org.
Turing AM. On computable numbers, with an application to the Entscheidungsproblem. J. Math 1936;58(345–363):5.
Turner P. Selective and intelligent imaging using digital evidence bags. Digital Investigation 2006;3:59–64.
X-Ways. X-ways forensics. 2015. http://www.x-ways.net/forensics/.