Towards Collaborative Forensics: Preliminary Framework

Mike Mabey and Gail-Joon Ahn
Laboratory of Security Engineering for Future Computing (SEFCOM)
Arizona State University, Tempe, AZ 85281, USA
{mmabey, gahn}@asu.edu

Abstract

Digital forensic analysis techniques have been significantly improved and evolved in past decade but we still face a lack of effective forensic analysis tools to tackle diverse incidents caused by emerging technologies and the advances in cyber crime. In this paper, we propose a comprehensive framework to address the efficacious deficiencies of current practices in digital forensics. Our framework, called Collaborative Forensic Framework (CUFF), provides scalable forensic services for practitioners who are from different organizations and have diverse forensic skills. In other words, our framework helps forensic practitioners collaborate with each other, instead of learning and struggling with new forensic techniques. Also, CUFF uses and augments current and emerging standards, including DFXML and EDRM XML for concise file representation and efficient resource transmission. In addition, we describe fundamental building blocks for our framework and corresponding system requirements.

1. Introduction

Computer crime has swiftly evolved into organized, and in some cases state-sponsored, cyber warfare. The tools digital forensic examiners currently use are too limited to take on the challenges that are rapidly approaching their forensic cases. Before long, fundamental changes in the industry will make many of the forensic techniques used today obsolete [11]. Although many contributing elements can be identified, the heart of the problem is that current digital forensic examinations are too time-inefficient. The three principal causes of this inefficiency are summarized as follows:

Software Limitations: Single workstation computers have served as the primary tool of our society's computing needs for a long time. With the evidence data sets being as large as they are, a single computer simply does not have the resources to deliver analysis results in a timely manner.

Size of Evidence Data: Today a 1TB hard drive can be purchased for about $60 and the average hard drive cost per GB is less than $0.10 [2]. Such low cost makes terabyte-sized systems commonplace among even non-tech-savvy consumers. With such a proliferation of huge storage systems filled with user data, examiners are up against a mountain of stored data to work through [17]. The problem is compounded when the situation involves a RAID [20] or network attached storage unit shared among individuals or employees.

Increased Examiner Workload: As if insufficient tools and large datasets were not enough, digital crime continues to increase in popularity [14, 12, 15], naturally resulting in more investigations. Furthermore, state-sponsored cyberwar promotes the development of increasingly sophisticated software. Simply trying to keep up with the latest methods of penetration, exfiltration, and attack is insufficient to accommodate the pace of digital crime.

In addition, when cases become backlogged, only those designated as more urgent are worked on, potentially leaving suspects' co-conspirators at large, capable of making more victims out of innocent people.

1.1. Motivation

The challenges above can be greatly reduced by a secure and robust infrastructure that facilitates *collaborative forensics* [16, 19], which we define as the willful cooperation between two or more forensic examiners during any step in the forensics process, for the benefit of sharing specialized knowledge, insight, experience, or tools. Two advantages of collaboration are of particular interest to us. First, collaboration allows people to draw from others' expertise, which is invaluable when working on problems of a diverse nature or when the problem set of a job constantly changes. Second, collaboration is a method of spreading a workload, which results in less time needed for the job to be completed.

IEEE IRI 2011, August 3-5, 2011, Las Vegas, Nevada, USA
978-1-4577-0966-1/11/$26.00 ©2011 IEEE

Consider the following hypothetical scenario. Past crimes committed by members of a white supremacists group have been contained to one state. However, with a recent expansion of operations to include a higher level of digital organization and recruiting, the group has begun to spread its activities across multiple states. As such, further investigation efforts may call for a combination of state and federal efforts, where previously only state enforcement was involved. These circumstances result in case files being stored in multiple locations by multiple agencies, which adds a new layer of complexity to sharing and cross-referencing critical information relating to this group.

Even in more localized investigation cases, evidence seizure may yield a variety of digital evidence, such as a mix of Windows, Linux, and Mac computers, cell phones, GPS devices, gaming consoles, et cetera. Since examiners must be certified to work on a particular type of evidence (depending on the investigating agency), such a workload must be split up among personnel. Furthermore, because there is no tool which can accommodate all evidence types, the evidence presentation lacks uniformity in format and structure.

While many generic collaboration solutions exist today, none of them have been crafted specifically for the needs of the digital forensics industry. To be truly effective, a collaborative forensics infrastructure should maintain the strict privacy and integrity principles the discipline demands, while also giving examiners the flexibility to communicate however is best for the situation. This demands a level of robustness that is simply not offered by collaboration tools at present.

Beyond just communication, collaboration also implies a sharing of resources. For a proper exchange of data (whether it be files needing to be analyzed or the results of an analysis), there must first be a uniform representation of that data, and then a common storage space solution where all collaborators can keep their resources secure. This will require the establishment of standards to ensure that all parties can access and interpret the data. Means to efficiently manage resources will also be needed.

If examiners are to collaborate on a large scale, it will also be crucial for this infrastructure to provide vast amounts of computing power, which is best accomplished through some distributed processing method. Ideally, a distributed processing solution would also include scalable resources. Because there is not a single technological solution that will properly meet this need for all organizations, there must be a generic way to interface for such processing resources.

To best facilitate collaboration among examiners, a collaborative forensics solution should not be limited to supporting its use on a small number of operating systems. This would hinder the collaboration process and may exclude experts who could offer potentially crucial insight.

In addition, as forensic analysis and presentation methods evolve, examiners need to incorporate these methods into their digital forensics software tool.

The rest of this paper is organized as follows. We first discuss the progress made by others in related fields in Section 2. In Section 3 we provide the architecture of our solution, which is an abstraction of the most essential components. We then discuss a realization of our framework in Section 4 which introduces all other necessary components. Section 5 concludes this paper.

2. Related Work

Many efforts have been made to improve the efficiency and versatility of digital forensic tools. Roussev and Richard proposed a method for moving away from single workstation processing for forensic examination to a distributed environment [17]. Liebrock et al. proposed improvements upon Roussev and Richard's system in [13], which introduced a decoupled front-end to a parallel analysis machine.

In [18], Scanlon and Kechadi introduced a method for remotely acquiring forensic copies of suspect evidence which transfers the contents of a drive over a secure Internet connection to a central evidence server. While this effort is a step for the better in terms of making evidence centrally accessible, it is difficult to see the direct utility of such an approach without accompanying software or analysis techniques to take advantage of storing the evidence on a server.

Other research efforts have focused on specific obstacles in the forensic examination process. For example, Urias and Liebrock reported on issues encountered when attempting to use a parallel analysis system on RAID storage systems [20]. Similarly, methods of properly handling the challenges presented by encrypted drives have been presented by Casey and Stellatos in [6] and by Altheide et al. in [5].

Garfinkel has made great efforts to create standards to improve the overall digital forensic examination process. Garfinkel et al. presented the details of a forensic corpora in [10] with the purpose of giving researchers a systematic way to measure and test their tools. Garfinkel took this a step further in [8] with his work to represent file system metadata with XML. Finally, in [11] Garfinkel put forth a challenge to researchers and developers everywhere to take note of the current industry trends and take them head on with innovative forensic solutions that match the properties of emerging technologies.

Since our realization of our framework is built upon a cloud, we also consider work done by researchers to address some of the issues related to shared storage in a cloud. Du et al. proposed an availability prediction scheme for sharable objects, such as data files or software components, for multi-tenanted systems in [7]. In [22], Wang et al. introduced a middleware solution to improve shared IO performance with Amazon Web Services [1].

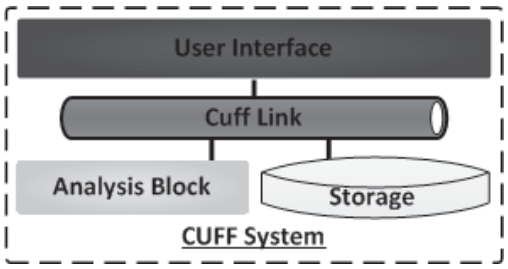
There also appears to be a trend toward supporting collaboration mechanisms in digital forensics tools such as FTK 3 [3] But to the best of our knowledge, there has yet to be a single system which can satisfy all the functionalities set forth in Section 1 in a truly robust manner.

3. CUFF: Collaborative Forensic Framework

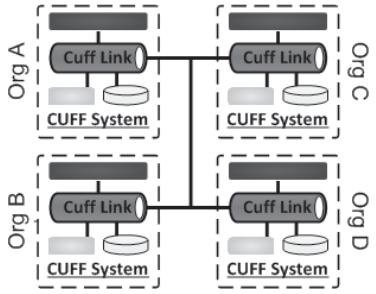
Based on the features and requirements necessary to achieve collaborative forensics as enumerated in Section 1, this section describes our framework, called Collaborative Forensic Framework (CUFF), and elaborates what mechanisms are needed to facilitate these features. Our framework consists of four core components (i) to mediate communication between components in the system, (ii) to coordinate the distributed analysis processing, (iii) to maintain the shared storage space, and (iv) to provide a basic interface to the system for the user interface. While a precise set of APIs for these four components may vary for the deployment setting, they should always fulfill specific foundational operations and always have the same basic interactions with the other components. We now discuss these two points in context of each component. Figure 1 gives an overview of how these components relate to each other.

The Analysis Block is the workhorse of the system, and in truth all other components are simply in place to either provide an interface to it, or to facilitate its proper function. The Analysis Block includes an Analysis Block Controller (ABC) as well as all processing resources. Ideally, the processing resources would be quite substantial and capable of handling a continuous inflow of analysis jobs of significant size. The ABC will receive a large number of analysis requests, and is expected to enqueue and dequeue each job request in an organized and efficient manner, which should also be fault-tolerant and maintain a high level of responsiveness. Because it is in charge of maintaining the queue of jobs, the ABC oversees the processing resources and ensures that they are used properly.

When assigning processing resources a job, the ABC should take into account both the required computational resources of the job (which can be computed internally) as well as the criticality level of the case with



(a) The Cuff Link provides the means for inter-system communication.



(b) Multiple deployments of CUFF can communicate with each other through the Cuff Link

Figure 1. Overview of CUFF.

which the job is associated (which would need to be entered by the user). For example, if a high-profile case is opened and input to the system, or if a data set requires analysis by an algorithm that takes a particularly long time to complete, the jobs associated with these tasks would be placed closer to the front of the queue. However, the ABC must be flexible enough to be able to revise its initial assessment of a job's placement if, for example, a user demotes the level of criticality of a job based on evolving circumstances regarding the case.

The storage component keeps track of all acquired disk images, the analyses of their contents, comments and notes from users, and related information all need to be kept for performing forensic tasks. To do this, it must accept incoming data streams of acquired disk images, and strictly maintains the integrity of the data through validation of the original checksums.

Requests will come at a high rate from the processing resources in the Analysis Block, so the storage component's response time needs to be controlled. Data such as analysis results, user comments, and communications between users will need their own distinct class method for transfering to the storage component.

In coordination with whatever access control mechanism is implemented, the storage component also maintains strict confidentiality of the data it stores. The storage component must also be flexible enough to allow

temporary and/or limited access to case data for consulting professionals, allowing them to collaborate with those directly responsible for the case.

The storage component additionally uses an established standard for uniformly representing the structure of acquired images. It supports the transmission of data segments between components in the system, as well as the transmission between distinct CUFF-enabled systems. The standard also enables to verify the evidence representation.

The user interface is the access portal to the entire system. All features implemented in the system are closely coupled with the interface. The first and most essential of forensic operations is the acquisition of disk images for their storage in the system. The user interface supports the evidence acquisition and also is responsible for providing the means for users to communicate and share data and information with each other.

The Cuff Link mediates communication between all other components in the system. It validates parameter input and stores location information for each of the other components. Also, since it is the component that manages the forensic process, it is responsible for assigning examiners jobs and notifying supervisors when the work on a case has been completed. The Cuff Link maintains order in the system by dictating the available APIs for each of the other components. It also simplifies the implementation of other components by reducing the number of connections they must make down to one.

4. Realization of CUFF

In this section we describe our efforts at taking CUFF from an abstraction to a usable implementation upon which mechanisms can be built to make the system ready for practical use.

As we stated earlier,the most important component in CUFF is the Cuff Link, which is more than just a relay for the system. The Cuff Link acts as an intelligent forensics process manager, and is responsible for initiating deterministic events based on the progress in the case examination. This serves as a method of (i) reducing examiner overhead and improving communication by automating the workflow of the examination, and (ii) ensuring applicable rules of evidence are maintained, such as tracking the chain of custody.

In addition, it is required that all inter-component communication in the system must go through the Cuff Link. This is to be standardized and regulated through the use of agents which run on all nodes in the system. Considering that the greatest diversity in CUFF originates from the analysis nodes in the Analysis Block, it is most important that the agents on these nodes be well-designed. While all node agents will be programmed

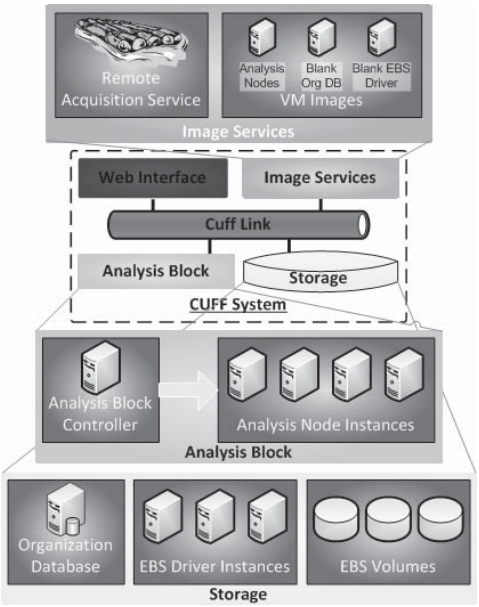


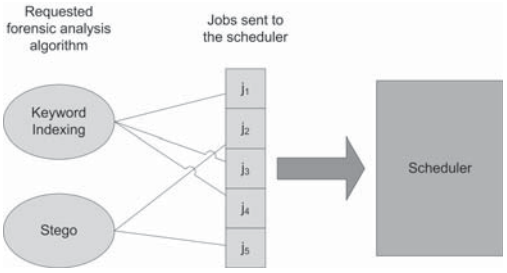
Figure 2. CUFF component details

with a standard set of communication protocols, each distinct analysis node's agent will be customized to the analysis programs being hosted on that node type. This allows the agent to store whatever parameters necessary to interface with the programs as well as retrieve the analysis results. Because much of the implementation for these nodes will be the same for all node types, this improves the ability to support new file systems, operating systems, analysis algorithms, and so forth.

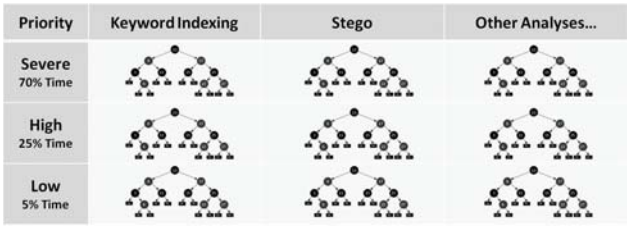
To store all the analysis node images which have been custom made to provide a certain set of analysis algorithms, we created the VM Image Storage subcomponent to act as a central storage location for the CUFF cloud. Because analysis node images are virtual machines, they can be configured to run a wide selection of operating systems, maximizing platform compatibility.

To further facilitate distributed analysis processing in CUFF, we have adopted the use of Garfinkel's Digital Forensics XML (DFXML) representation for file system metadata [8, 9]. DFXML gives researchers and developers a standard way of representing and accessing the contents of an imaged drive by storing the addresses and lengths of all "byte runs" (file fragments) on a disk. The DFXML file can then be used with the disk image for accessing specific files.

To provide another method of acquisition to examiners, we decided to add another component that would be exposed directly to the user, the Image Services component, which encapsulates two subcomponents, the Remote Acquisition Service and the VM Image storage.



(a) Jobs are first separated by the type of requested analysis algorithm when received by the ABC scheduler.



(b) Each algorithm type and priority has its own red-black tree. Severe jobs will be requested by analysis nodes 70% of the time, high jobs 25%, and low 5%.

Figure 3. Initial scheduling of jobs as they are received by the scheduler

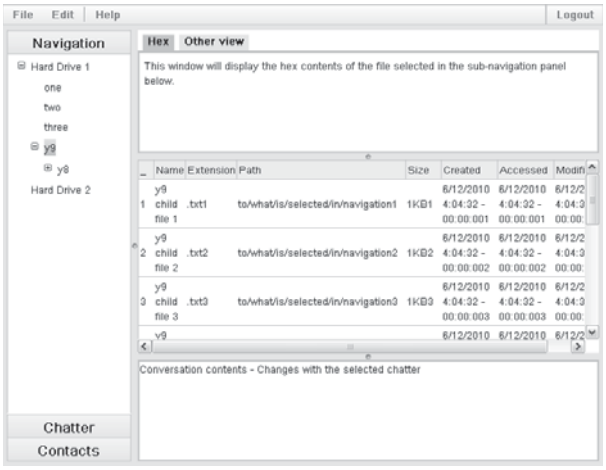


Figure 4. Simple web interface for CUFF.

Figure 2 illustrates the details of CUFF components.

In order for the Storage component of CUFF to work effectively in a cloud, we have chosen to use an elastic block store (EBS) functionality. In order to use EBS volumes, there must be a driver for the block to be attached to. Once again, these drivers will have an agent running on them to facilitate data transfer with other nodes, giving a generic interface to use. To facilitate the proper use of EBS volumes, the ABC has an additional subcomponent for resizing EBS volumes when needed, since a size must be specified at the EBS volume creation which may not be able to accommodate future storage needs.

To provide further details of how the ABC handles the prioritization and scheduling of analysis jobs requested by examiners, we first make the assumptions that (i) the user interface is designed to allow criticality level input, (ii) an analysis request (or *job*) will be a set of tuples consisting of a criticality level, a requested analysis algorithm, and a URI for the file, and (iii) analysis nodes are to be assigned a single file to analyze at once.

When a set of unscheduled jobs ($j_u = \{j_{u_i} | j_{u_i} = \langle r_{u_i}, a_{u_i}, f_{u_i} \rangle\}$) is sent to the ABC scheduler, S, they are separated by their requested algorithm, a_{u_i} (see Figure 3(a)), and by their user-specified priority, $r_{u_i} \in \{low, high, severe\}$, into different levels. f_{u_i} is the file's URI. Once separated, each subdivision of the original job set j_u is inserted into a red-black tree specific to the subdivision type. Jobs in the red-black trees are ordered by their arrival time as shown in 3(b).

For our initial implementation of a CUFF system, we have used the Google Web Toolkit to create a simple web interface. Our initial prototype is shown in Figure 4. In this prototype, we have included an area for text and voice conversations as well as a list of the user's contacts within the system. In the Viewing pane, we have left space for multiple data viewing tools to be added on to the system.

We also adopted the Google Wave Operational Transformation algorithm [21] for CUFF's user interface. This gives users the feeling of fluid communication, because even though they are communicating through text, they are able to respond more quickly to other users' comments because they are reading what others are typing while it is still being typed. More importantly than any other feature, our realization accommodates the main tasks of any digital forensic investigation: *Acquisition*: We have designed our system to be flexible with how the acquisition of evidence is performed. Examiners may either use the upload tool from the common web interface while in the lab, or they may initiate the upload process from a remote site via the Remote Acquisition Service; *Validation*: Our system guarantees the integrity of files from the earliest stages of the investigation through the use of hash values for every device image and file before and after every data transmission; *Discrimination*: Our system supports the use of discrimination and filter tools, which can use sets of hashes of known good files, such as the Reference Data Set provided by the National Institute of Standards and Technology [4], to highlight those files which are unknown, effectively eliminating an extensive number of files the examiner needs to look at; *Extraction*: As the

task which is typically most demanding of examiners' time, our web interface implementation has focused on making the features which support extraction as robust as possible; and *Reporting*: Comparable to contemporary tools, our system allows for users to generate reports of their findings with comments, as well as maintaining a strict log of activities any users performed on any of the evidence.

With our realization of CUFF, we have provided a means of executing the fundamental operations of the framework.

5. Conclusion

In this paper we have discussed the trends of computer crime and the tools to combat those crimes. From these trends we have determined that collaboration among examiners through a secure and robust system would give them a significant advantage to successfully identify both inculpatory and exculpatory evidence in a timely manner. We set forth our requirements for such a system in a framework based on principles of scalability and interoperability. We then described our implementation of the framework and the additional components that were necessary for the basic operations of a live deployment of CUFF. As we move forward in this research effort, we will focus on refining our approaches and overcoming the limitations we currently face.

References

[1] Amazon web services. `http://aws.amazon.com/`.

[2] Cost of hard drive storage space. `http://ns1758.ca/winch/winchest.html`.

[3] Forensic toolkit (ftk). `http://accessdata.com`.

[4] National software reference library. `http://www.nsrl.nist.gov/Downloads.htm`.

[5] C. Altheide, C. Merloni, and S. Zanero. A methodology for the repeatable forensic analysis of encrypted drives. In *EUROSEC '08: Proceedings of the 1st European Workshop on System Security*, pages 22–26, New York, NY, USA, 2008. ACM.

[6] E. Casey and G. J. Stellatos. The impact of full disk encryption on digital forensics. *SIGOPS Oper. Syst. Rev.*, 42(3):93–98, 2008.

[7] J. Du, X. Gu, and D. S. Reeves. Highly available component sharing in large-scale multi-tenant cloud systems. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, HPDC '10, pages 85–94, New York, NY, USA, 2010. ACM.

[8] S. Garfinkel. Automating disk forensic processing with sleuthkit, xml and python. In *IEEE Systematic Approaches to Digital Forensics Engineering*, pages 73 – 84, May 2009.

[9] S. Garfinkel. Aff and aff4: Where we are, where we are going, and why it matters to you. In *Sleuth Kit and Open Source Digital Forensics Conference*, 2010.

[10] S. Garfinkel, P. Farrell, V. Roussev, and G. Dinolt. Bringing science to digital forensics with standardized forensic corpora. *Digital Investigation*, 6(Supplement 1):S2–S11, 2009. The Proceedings of the Ninth Annual DFRWS Conference.

[11] S. L. Garfinkel. Digital forensics research: The next 10 years. *Digital Investigation*, 7(Supplement 1):S64 – S73, 2010. The Proceedings of the Tenth Annual DFRWS Conference.

[12] K. J. Higgins. Zeus attackers deploy honeypot against researchers, competitors. *DarkReading*, November 2010.

[13] L. M. Liebrock, N. Marrero, D. P. Burton, R. Prine, E. Cornelius, M. Shakamuri, and V. Urias. A preliminary design for digital forensics analysis of terabyte size data sets. In *SAC '07: Proceedings of the 2007 ACM symposium on Applied computing*, pages 190–191, New York, NY, USA, 2007. ACM.

[14] J. Menn. *Fatal System Error: The Hunt for the New Crime Lords Who are Bringing Down the Internet*. PublicAffairs, first edition, 2010.

[15] J. Menn. U.s. experts close in on google hackers. `http://www.cnn.com/2010/BUSINESS/02/21/google.hackers/index.html`, February 2010.

[16] L. Moraski. Cybercrime knows no borders. *Infosecurity*, `http://www.infosecurity-us.com/view/18074/cybercrime-knows-no-borders-/`, May 2011.

[17] V. Roussev and G. G. R. III. Breaking the performance wall: The case for distributed digital forensics. In *The Proceedings of the Fourth Annual DFRWS Conference*, 2004.

[18] M. Scanlon and M.-T. Kechadi. *Online Acquisition of Digital Forensic Evidence*, volume Volume 31 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 122–131. Springer Berlin Heidelberg, 2009.

[19] $(ISC)^2$ U.S. Government Advisory Board Executive Writer's Bureau. Do punishments fit the cybercrime? *Infosecurity*, `http://www.infosecurity-us.com/view/12029/do-punishments-fit-the-cybercrime-/`, August 2010.

[20] V. Urias, C. Hash, and L. M. Liebrock. Consideration of issues for parallel digital forensics of raid systems. *Journal of Digital Forensic Practice*, 2008.

[21] D. Wang, A. Mah, and S. Lassen. Google wave operational transformation. `http://wave-protocol.googlecode.com/hg/whitepapers/operational-transform/operational-transform.html`, July 2010. Version 1.1.

[22] J. Wang, P. Varman, and C. Xie. Middleware enabled data sharing on cloud storage services. In *Proceedings of the 5th International Workshop on Middleware for Service Oriented Computing*, MW4SOC '10, pages 33–38, New York, NY, USA, 2010. ACM.