Performance of a Logical, Five- Phase, Multithreaded, Bootable Triage Tool

Ibrahim Baggili, Andrew Marrington, Yasser Jafar

Chapter 19

PERFORMANCE OF A LOGICAL, FIVE-PHASE, MULTITHREADED, BOOTABLE TRIAGE TOOL

Ibrahim Baggili, Andrew Marrington and Yasser Jafar

Abstract This paper describes a five-phase, multi-threaded bootable approach to digital forensic triage, which is implemented in a product called Forensics2020. The first phase collects metadata for every logical file on the hard drive of a computer system. The second phase collects EXIF camera data from each image found on the hard drive. The third phase analyzes and categorizes each file based on its header information. The fourth phase parses each executable file to provide a complete audit of the software applications on the system; a signature is generated for every executable file, which is later checked against a threat detection database. The fifth and final phase hashes each file and records its hash value. All five phases are performed in the background while the first responder interacts with the system. This paper assesses the forensic soundness of Forensics2020. The tool makes certain changes to a hard drive that are similar to those made by other bootable forensic examination environments, although the changes are greater in number. The paper also describes the lessons learned from developing Forensics2020, which can help guide the development of other forensic triage tools.

Keywords: Triage tool, bootable tool, forensic soundness, performance

1. Introduction

Digital forensic triage is a topic of considerable interest to the digital forensics community. In the medical realm, three conditions must be satisfied for triage to occur [7]: (i) a scarcity of health resources; (ii) a triage officer must be able to assess the medical needs of each patient based on a brief examination; and (iii) the triage officer must be able to apply an algorithm or criteria to determine the treatment and treatment priority for each patient.

The practice of medical triage arose from emergency situations in wartime. The earliest medical triage systems date back to the nineteenth century; many scholars attribute the first formal battlefield triage system to Baron Dominique-Jean Larrey, chief surgeon of Napoleon's Imperial Guard. Before medical triage came into existence, wounded soldiers were not treated effectively, sometimes not at all. Soldiers typically helped each other; there was no systematic approach for specialists to provide medical attention, creating a backlog of wounded soldiers.

The same situation frequently occurs when dealing with computer crimes. In the traditional digital forensic model, a computer system is seized and is typically transported to a laboratory where it is forensically imaged and analyzed. The vast majority of digital forensic practitioners state that they have huge case backlogs; often, the processing delays are as much as eight to twelve months [9]. The same problem is encountered with mobile devices [10]. Delays in the forensic laboratory translate to delays in criminal investigations and court proceedings, resulting in deleterious effects on the course of justice [4]. The digital forensic case backlogs have prompted research on systematic methodologies for improving the digital forensic process and, more specifically, digital forensic triage models and procedures.

2. Related Work

A number of traditional models have been developed for digital forensic investigations [2, 3, 12, 15]. The traditional models involve a number of phases that focus on collecting evidence from a crime scene and processing the evidence in a laboratory environment [11]. These models have influenced the development of digital forensic triage models, primarily by emphasizing the idea of conducting an early inspection of a computer system at a crime scene before bringing it back to the laboratory.

Adelstein [1] has proposed the use of a mobile forensic platform to facilitate triage without requiring evidence to be brought back to the laboratory. Richards and Roussev [13] describe a a conceptually similar triage approach that they call "on-the-spot" forensics. Both approaches demonstrate that significant advantages can be gained during an investigation in terms of the reduction in time and the possibility of obtaining confessions from suspects when inspecting computer systems at the crime scene.

Rogers, *et al.* [14] have proposed a formal model for digital forensic triage. Their Cyber Forensics Field Triage Process Model (CFFTPM) incorporates an on-site field approach to digital forensic investigations. In particular, the model seeks to: (i) find usable evidence immediately;

(ii) identify the victims at acute risk; (iii) guide the ongoing investigation; (iv) identify the potential charges; and (v) accurately assess the offender's danger to society while preserving the integrity of the evidence for further examination and analysis. Much of the analysis can be performed on scene, where the investigator can generate user profiles, examine the home directory, file properties and registry, create a chronology and search for browser artifacts, email and instant messages [14]. However, one of the main drawbacks of CFFTPM is that it is highly manual and can only be performed by well-trained and experienced digital forensic investigators who can leverage a broad set of forensic tools.

Casey, *et al.* [4] propose a three-tiered approach to digital forensic examinations: (i) survey/triage forensic inspection, during which the potential sources of digital evidence are identified; (ii) preliminary forensic examination, during which relevant digital evidence is interpreted for individuals involved in the case (e.g., to assist in interviewing a suspect); and (iii) in-depth forensic examination, during which a full examination (typically, manual and time intensive) is performed to recover all the available digital evidence. The first two stages, triage and preliminary forensic examination, are intended to assist the investigation. The third stage is primarily intended to support the case going to trial [4]. Most of the well-known digital forensic tools support the third stage, although they may be used to support the other stages (especially the preliminary forensic examination stage).

3. Forensics2020

The Forensics2020 tool described in this paper was developed as a result of an academia-industry partnership involving the first author and Cryptic Solutions (`www.cryptic.co.uk`). The tool was built after numerous meetings with digital forensic investigators in Ireland, the United Arab Emirates, the United Kingdom and the United States. During these meetings, law enforcement officers and corporate forensic investigators discussed the challenges they faced with existing tools and the lack of effective, easy-to-use digital forensic triage tools.

The law enforcement officers and corporate investigators were very interested in a digital forensic triage system that could deployed easily. It was highly desirable that the system could be operated by non-experts with minimal training (including first responders at crime scenes). Other important features included an automated system with minimal configuration and input, and the ability to view results while the system was processing evidence. Finally, it was deemed critical that the triage system operate in a forensically sound manner and always maintain the integrity of the evidence.

These recommendations were used as guidelines when designing and implementing the Forensics2020 tool. The CFFTPM was selected as the underlying framework due to its wide adoption by U.S. law enforcement agencies.

3.1 Software Design

The Forensics2020 triage tool was designed to serve as a bootable environment. A bootable environment is a controlled environment where the triage software runs in a known-safe operating system as opposed to running live in the uncontrolled environment of a native operating system. This enhances the trustworthiness of the results; the known-safe operating system is free from malware and anti-forensic software that may be present in the suspect's system. The bootable forensic examination environment approach is also common to many forensic live CD distributions (e.g., Helix3 Pro) and, thus, has an accepted basis.

However, there are two significant drawbacks to a bootable approach. First, items in memory could potentially be lost during the process of booting into the forensic triage environment. Second, if whole disk encryption is used, then triage will not be possible unless the encryption keys are known to the first responder. A live examination could address both these problems, but it could be subject to interference by malware and other untrusted software running on the suspect's computer system.

The Forensics2020 software is multi-phased and multi-threaded. The software is loaded from a bootable Windows Pre-installation Environment (PE) using a USB stick. The triage process is initiated after a password is entered. At this point, the first responder has the option to un-check a triage phase if it is not to be performed.

Forensics2020 has five major automated phases that proceed in sequence. The phases are illustrated in Figure 1 and their details are provided in Table 1. The order of the phases was chosen based on the time it typically takes for each phase to be completed – the first phase is the fastest and the last phase is the slowest (see Figure 5).

Since the Forensics2020 phases are threaded away from the user interface, the first responder can introduce other evidence via the interface and view it while the five triage phases complete in the background. At any time, the first responder can click on any part of the user interface and interact with the tool to view important evidentiary items such as registry keys, web browser history, and iPhone and Blackberry backup structures.

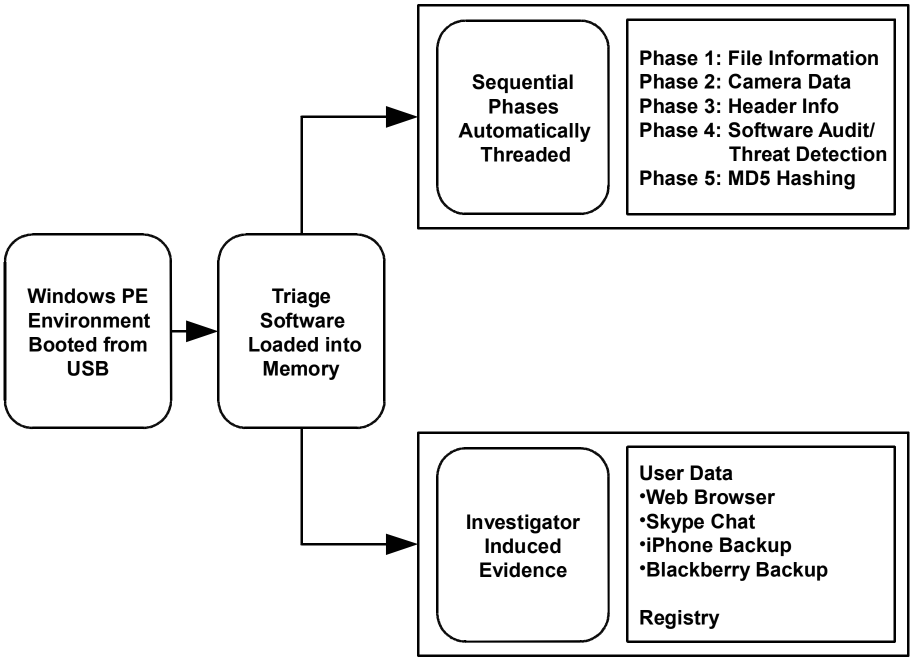
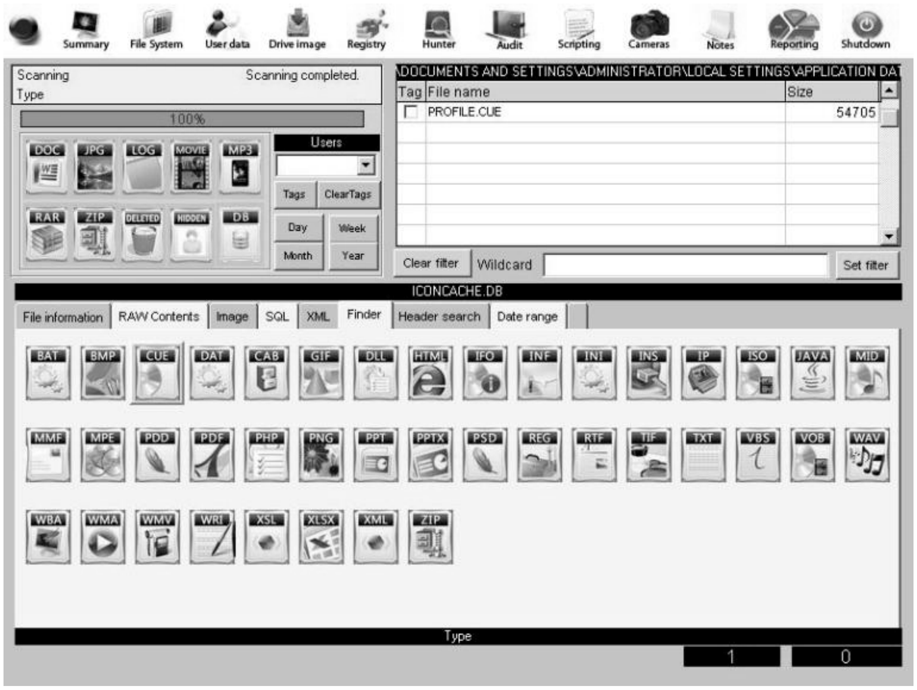

Windows PE Environment Booted from USB
Triage Software Loaded into Memory
Sequential Phases Automatically Threaded
Phase 1: File Information
Phase 2: Camera Data
Phase 3: Header Info
Phase 4: Software Audit/ Threat Detection
Phase 5: MD5 Hashing
Investigator Induced Evidence
User Data
•Web Browser
•Skype Chat
•iPhone Backup
•Blackberry Backup
Registry

Figure 1. High-level Forensics2020 triage process.

Table 1. Forensics2020 phases.

Phase	Description
Phase I: File Information	Every file is logically collected from the hard drive along with its file metadata.
Phase II: Camera Data	Every photograph is analyzed for EXIF data.
Phase III: Header Information	Every file is analyzed and categorized based on its header, not on its extension.
Phase IV: Software Audit/ Threat Detection	The EXE files are parsed to provide a complete audit of all the software applications on the system. A proprietary algorithm is applied to each of the files to generate a signature. The signature is compared against a threat database if the user chooses to initiate this function.
Phase V: MD5 Hashing	Every file is hashed and the hash value is recorded. The hash values may be used to: (i) verify data integrity; and (ii) check against a hash value dataset that is relevant to the investigation.



Scanning completed.

\DOCUMENTS AND SETTINGS\ADMINISTRATOR\LOCAL SETTINGS\APPLICATION DAT

Tag	File name	Size
□	PROFILE.CUE	54705

Figure 2. Forensics2020 screen functionality.

3.2 Using Forensics2020

While Forensics2020 is operating, the first responder can interact with the user interface to see the results to that point and to request certain types of data. The first responder can search for file types based on their file extensions after Phase I is completed, and can search for file headers after Phase III is complete. The first responder can also filter searches based on files modified, accessed or created on specific dates or within specific time ranges, and can view files in raw format or in a "preview" mode (for supported file types). Figure 2 shows an example Forensics2020 screen that enables a first responder to search for specific types of files.

The first responder can also direct Forensics2020 to gather and view certain types of data from the filesystem. The data includes web browser history, Skype chat logs, iOS and Blackberry backup files, and other data relevant to the case. The first responder can also have the Windows registry mounted and search for particular keys that could reveal hardware devices that were connected to the system and software that was installed on the system.

4. Experiments

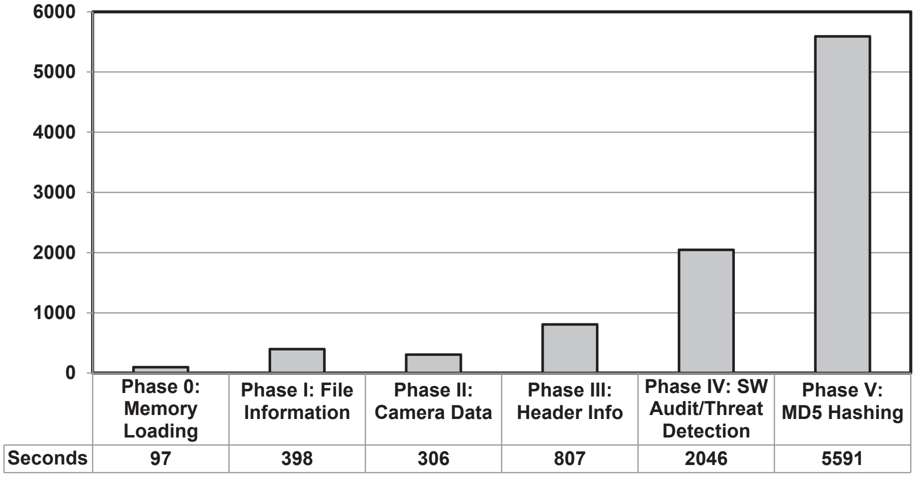
We conducted five experiments to evaluate the efficacy of the Forensics2020 triage tool. The first experiment focused on testing the performance of Forensics2020 running on personal computers located in several university computer laboratories ($n = 57$). Four additional experiments were then conducted to validate the integrity of the hard drives and the files after Forensics2020 was loaded.

4.1 Experiment 1

The goal of this experiment was to test the Forensics2020 triage procedure on real computer systems that had been used for general purpose computing instead of computer systems or virtual machines (VMs) with traces of computer activity that were contrived specifically as an experimental dataset. The computers examined in the experiment were physical machines (not VMs) in laboratories that were made available to students for use in their classes, although the students may also have used the computers for personal reasons and for activities only peripherally related to their studies. Each computer may have had several "regular users" who consistently used the computer for their classwork, in addition to dozens of casual users who may have used the computer between classes for any purpose. A computer located in a shared computer laboratory typically has a different usage pattern compared with a computer located at an individual's home. Nevertheless, the computers used in the experiment were real computers with traces of real user activity. As such, they represented the most realistic dataset available for the experiment.

Overall, 26.33 TB of data was analyzed in the experiment; the average hard drive size was 473 GB. To assess the performance of Forensics2020, the tool was run on the test computers and the following times were recorded for each computer: (i) time taken for the tool to load into memory; and (ii) time taken for each of the five phases to complete.

- **Performance Results:** Figure 3 shows the average time (in seconds) for each of the five phases across the entire dataset. It is important to note that Phase II is faster than Phase I because the number of digital images triaged was significantly lower than the number of files found on the computer systems. Other interesting findings are:
  - Four computers generated errors during or after the scans.
  - Triage failed on one computer due to the condition of the hard drive (1.75% failure rate).



	Phase 0: Memory Loading	Phase I: File Information	Phase II: Camera Data	Phase III: Header Info	Phase IV: SW Audit/Threat Detection	Phase V: MD5 Hashing
Seconds	97	398	306	807	2046	5591

Figure 3. Average execution time of each phase across all 57 computers.

- Three failures (5.26% failure rate) were observed during Phase V (MD5 hashing).
- The average time taken to complete all five phases was 5,591 seconds (1 hour, 33 minutes and 11 seconds).
- The maximum time taken to complete all five phases was 10,356 seconds (2 hours, 52 minutes and 36 seconds).

■ **User Data Results:** After all the computers were triaged, the results from every triage report were recorded and the data was analyzed accordingly. The following is a summary of the user data results for the 57 computers:

- 260,101 software audit entries were detected on the 57 computers.
- 40 software audit entries across the 57 computers were identified as possible viruses or malware (0.70 infections per computer).
- On the average, 4,563 software audit entries were detected per computer.
- Only one iPod backup was found.
- 525 browsing history entries were detected for Google Chrome on the 57 computers (average of 9.21 entries per computer).
- 351 browsing history entries were detected for Mozilla Firefox on the 57 computers (average of 6.15 entries per computer).

- 1,408 browsing history entries were detected for Internet Explorer on the 57 computers (average of 24.7 entries per computer).
- 3,475 Microsoft Word and Microsoft Excel files were found on the 57 computers (average of 60.96 Word/Excel files per computer).
- On the average, 14 camera models were detected per computer.
- As many as 44 camera models were detected on one of the 57 computers and 150 photo EXIF details were linked to the same computer.
- More than 2,340 photograph EXIF details were detected on the 57 computers.

4.2 Experiment 2

The goal of this experiment was to test if Forensics2020 modified the hard drive of a scanned computer.

- **Equipment:**
  - A Windows XP laptop with a 160 GB SATA hard drive.
  - A Logicube Forensic Dossier hardware imaging and verification device.
  - A Forensics2020 bootable USB stick.

- **Procedure:**
  1. The hard drive was extracted from the laptop and hashed using the Forensic Dossier device.
  2. The Forensics2020 USB stick was inserted into the laptop.
  3. The laptop was forced to boot from the Forensics2020 USB stick.
  4. Forensics2020 was left running until all the automated phases were completed upon which Forensics2020 was shut down.
  5. The hard drive was extracted and hashed again using the Forensic Dossier device.
  6. The hash values for the hard drive before and after running Forensics2020 were compared.

Table 2. Experiment 2 results.

Stage	**Hash**
SHA-256 of the hard drive before running Forensics2020	D8C632E0 D922A8D9 6DCB17C1 3FE8A904 F821726E D405266D 736F0498 00885F3C
SHA-256 of the hard drive after running Forensics2020	D8C21317 E72B5577 0B40FA22 9B203744 FCE916AD 55017326 2B8f88E4 5824B4E6

- **Results:** Table 2 shows that different hash values were obtained for the hard drive before and after running Forensics2020. Additional experiments were required to investigate precisely when the changes occurred.

4.3 Experiment 3

The goal of this experiment was to investigate if the hash values changed after the Windows PE operating system was booted but before the laptop hard drive was mounted. The same equipment as in Experiment 2 was used.

- **Procedure:**

  1. The hard drive was extracted from the laptop and hashed using the Forensic Dossier device.
  2. The Forensics2020 USB stick was inserted into the laptop.
  3. The laptop was forced to boot from the Forensics2020 USB stick.
  4. Forensics2020 was shut down after the Windows PE operating system was loaded, but before the triage process was started (right before the program splash screen). Thus, the hard drive was not yet mounted when Forensics2020 was shut down.
  5. The hard drive was extracted and hashed again using the Forensic Dossier device.
  6. The hash values for the hard drive before and after running Forensics2020 were compared.

- **Results:** As shown in Table 3, the hash values for the hard drive did not change during the experiment; the integrity of the hard drive was maintained. This means that loading Forensics2020 into memory did not cause any hard drive writes. An additional experiment was required to determine if the hash values were changed after the hard drive was mounted.

Table 3. Experiment 3 results.

Stage	**Hash**
SHA-256 of the hard drive before running Forensics2020	`55407B01 4F4847E0 66F25B87 FD24FBDC 73B4969B B94434E6 22CAFAE5 D72E2C17`
SHA-256 of the hard drive after starting Forensics2020 and aborting before the program splash screen loaded	`55407B01 4F4847E0 66F25B87 FD24FBDC 73B4969B B94434E6 22CAFAE5 D72E2C17`

4.4 Experiment 4

The goal of this experiment was to determine if the hard drive retained its integrity after the Windows PE operating system was booted, after Forensics2020 was loaded into memory and after the laptop hard drive was mounted. The same equipment as in Experiment 2 was used.

- **Procedure:**

  1. The hard drive was extracted from the laptop and hashed using the Forensic Dossier device.
  2. The Forensics2020 USB stick was inserted into the laptop.
  3. The laptop was forced to boot from the Forensics2020 USB stick.
  4. Forensics2020 was shut down after the Windows PE operating system was loaded and after the triage process was started (right after the program splash screen). Thus, the hard drive was mounted when Forensics2020 was shut down.
  5. The hard drive was extracted and hashed again using the Forensic Dossier device.
  6. The hash values for the hard drive before and after running Forensics2020 were compared.

- **Results:** Table 4 shows the hash values for the hard drive before and after running Forensics2020. The results show that the hard drive was modified when it was mounted by the bootable Windows PE operating system. Therefore, it can be concluded that the changes were made to the hard drive as soon as it was mounted.

4.5 Experiment 5

In order to quantify the changes made by the Windows PE operating system mounting process, a differential analysis was performed on the

Table 4. Experiment 4 results.

Stage	**Hash**

SHA-256 of the hard drive before running Forensics2020	`55407B01 4F4847E0 66F25B87 FD24FBDC`
	`73B4969B B94434E6 22CAFAE5 D72E2C17`

SHA-256 of the hard drive after running Forensics2020, after the splash screen loaded and after the hard drive was mounted	`783C9989 94F7A6A2 103C70DE C6A419ED`
	`C2B7B748 84F8B886 7AB31AB5 E2739F17`

hard drive images taken before and after running Forensics2020. Differential analysis allows a forensic examiner to focus on the changes brought about as a result of the activity that takes place between the acquisition of an image A and the acquisition of an image B at a subsequent point in time [6].

- **Equipment:**

  – A Tableau T14 FireWire 800 IDE Bridge (write blocker).
  – A Windows XP "suspect" laptop with a 40 GB IDE hard drive.
  – A Windows 7 forensic workstation.
  – A Forensics2020 bootable USB stick.
  – AccessData FTK 1.71.
  – AccessData FTK Imager 3.1.2.0.

  Note that no images were acquired in the previous experiments. Instead, the hard drive was merely hashed before and after running Forensics2020. In this experiment, since differential forensic analysis had to be performed, it was necessary to acquire actual images of the "suspect" laptop for examination. Therefore, a write blocker was used instead of the Forensic Dossier device.

- **Procedure:**

  1. The hard drive was extracted from the "suspect" laptop.
  2. An image of the "suspect" hard drive was acquired using FTK Imager, employing the hardware write blocker to prevent inadvertent writes.
  3. The hard drive was replaced in the "suspect" laptop.
  4. The Forensics2020 USB stick was inserted into the "suspect" laptop.

5. The "suspect" laptop was forced to boot from the Forensics2020 USB stick.
6. The system was shut down after the Windows PE operating system was loaded and after Forensics2020 was started (right after the program splash screen). Thus, the hard drive was mounted when Forensics2020 was shut down.
7. The hard drive was extracted from the "suspect" laptop.
8. An image of the "suspect" hard drive was acquired using FTK Imager, employing the hardware write blocker to prevent inadvertent writes.
9. The two image files were examined using AccessData FTK. Using the file filter, known "ignorable" files, OLE files and duplicate files were excluded. Duplicates in FTK are files that already existed when they were discovered – not the files that exist on both images; consequently, it was necessary to sort by whether the item was "primary" as well as to identify only the files that were unique to a particular image or different in each image. This procedure yielded a list of files/items recovered from unallocated space where there were differences in the contents of the two hard drive images.

- **Results:** Consistent with the results of Experiments 3 and 4, the two image files had different hash values. Using the FTK file filter and sorting functionality, a differential analysis was performed on the two images. The following observations are significant:
  - The `$I30` file in each directory on the "suspect" hard drive was modified when the drive was mounted. `$I30` is the NTFS directory index file that employs a B-tree data structure instead of a flat file structure; Windows constantly rebalances the B-tree structures and, thus, modifies the `$I30` file. Interested readers are referred to [17] for a discussion of the forensic implications of `$I30` files.
  - The `$LogFile` was changed.
  - The `$MFT` was changed.
  - The `*.log` files in `C:\Windows\System32\config` were modified (to no significant extent).
  - The deleted files corresponding to Symantec Anti-Virus were changed.
  - Two deleted sets of Virtual PC `*.vhd` files were modified.

Table 5. Hard drive changes by bootable forensic examination environments.

Forensic Tool	**Suspected Alteration**	**Altered File Hashes**	**Files Altered by Search**

Helix3 Pro	Yes	Yes	`$LogFile`
			`$MFT`
			`C:\WINDOWS\bootstat.dat`

Kali	Yes	Yes	`C:\$I30`
			`$LogFile`
			`$MFT`
			`C:\WINDOWS\bootstat.dat`

Knoppix	Yes	Yes	`C:\$I30`
			`$MFT`
			`C:\Documents And Settings\ Administrator\Local Settings\ $I30`
			`C:\Program Files\Cisco Packet Tracer 5.3.1\sounds\ simulationTab.wav`

Forensics2020	Yes	Yes	1,667 different hash values

When considering the results of Experiment 5, it is important to note that many commonly-used Linux-based bootable forensic examination environments (e.g., Helix3 Pro and Backtrack) also perform mounts that modify filesystem metadata [16]. Therefore, the issues noted for Forensics2020 are not unique to the product or to the Windows PE operating system – they are problematic for other bootable forensic examination environments as well. Fathy, *et al.* [5] conducted an experiment similar to Experiment 5 for three bootable Linux-based forensic examination environments. Their results for Helix3 Pro, Kali and Knoppix are presented alongside the Forensics2020 results in Table 5. The Windows PE drive mounting process for Forensics2020 made the same sorts of changes as seen with Helix3 Pro, Kali and Knoppix, but in much greater quantities. However, the differential analysis revealed that Forensics2020 made no changes to the logical filesystem – the userspace files were not modified; only the filesystem metadata was changed. Whether or not this is acceptable from the perspective of forensic soundness is an open question.

5. Conclusions

The experimental results demonstrate that Forensics2020 is an effective digital forensic triage tool. On the average, files on the hard drive can be enumerated within 398 seconds; the slowest phase, however, is MD5 hashing. The semi-automated and multi-phase implementation enables a first responder to interact with Forensics2020 while digital forensic processes are executing in the background, giving the first responder more control over the investigation. Like other bootable forensic examination environments, Forensics2020 makes certain changes to the hard drive. Differential analysis revealed that Forensics2020 made no changes to userspace files; only the filesystem metadata was modified. However, this result is, nevertheless, a concern from the perspective of forensic soundness.

Future efforts will concentrate on extensive testing of Forensics2020 in real-world investigations. Also, experiments will be undertaken to measure if an automated tool like Forensics2020 actually reduces the time spent on digital forensic investigations [8]. With respect to forensic soundness, a sector-by-sector comparison of the hard drive before and after the triage process will be conducted to provide better insights into the nature of the changes to digital evidence.

Finally, our experience developing and deploying Forensics2020 have resulted in two key lessons learned with regard to digital forensic triage. First, a multi-threaded, multi-stage approach that enables first responders to interact with the evidence while the system performs its forensic processing is a desirable feature that enhances performance and saves time. The second lesson is that the manner in which the hard drive is mounted by a bootable triage tool is crucial to perceptions of forensic soundness. It could be that small and predictable modifications of a suspect's hard drive would have to be accepted in order to facilitate triage using a bootable examination environment, but the nature and causes of the modifications must be thoroughly documented if the evidence recovered and analyzed by Forensics2020 is to have probative value in court proceedings.

Acknowledgement

The authors wish to acknowledge the collaboration with Cryptic Solutions, especially with David Shearmon, CEO of Cryptic Solutions; and also with David Duke, the developer of Forensics2020.

References

[1] F. Adelstein, MFP: The Mobile Forensics Platform, *International Journal of Digital Evidence*, vol. 2(1), 2003.

[2] N. Beebe and J. Clark, A hierarchical, objectives-based framework for the digital investigations process, *Digital Investigation*, vol. 2(2), pp. 147–167, 2005.

[3] B. Carrier and E. Spafford, Getting physical with the digital investigation process, *International Journal of Digital Evidence*, vol. 2(2), 2003.

[4] E. Casey, M. Ferraro and L. Nguyen, Investigation delayed is justice denied: Proposals for expediting forensic examinations of digital evidence, *Journal of Forensic Sciences*, vol. 54(6), pp. 1353–1364, 2009.

[5] A. Fathy, A. Marrington, F. Iqbal and I. Baggili, Testing the forensic soundness of forensic examination environments on bootable media, submitted for publication, 2014.

[6] S. Garfinkel, A. Nelson and J. Young, A general strategy for differential forensic analysis, *Digital Investigation*, vol. 9(S), pp. S50–S59, 2012.

[7] K. Iserson and J. Moskop, Triage in medicine, Part I: Concept, history and types, *Annals of Emergency Medicine*, vol. 49(3), pp. 275–281, 2007.

[8] J. James, A. Lopez-Fernandez and P. Gladyshev, Measuring accuracy of automated investigation tools and procedures in digital investigations, presented at the *Fifth International Conference on Digital Forensics and Cyber Crime*, 2013.

[9] D. Kennedy and D. Sun, How to triage computer evidence: Tackling Moore's law with less, *Evidence Technology Magazine*, vol. 8(2), 2010.

[10] R. Mislan, E. Casey and G. Kessler, The growing need for on-scene triage of mobile devices, *Digital Investigation*, vol. 3(3-4), pp. 112–124, 2010.

[11] M. Pollitt, An ad hoc review of digital forensic models, *Proceedings of the Second International Workshop on Systematic Approaches to Digital Forensic Engineering*, pp. 43–54, 2007.

[12] M. Reith, C. Carr and G. Gunsch, An examination of digital forensic models, *International Journal of Digital Evidence*, vol. 1(3), 2002.

[13] G. Richard III and V. Roussev, Digital forensics tools: The next generation, in *Digital Crime and Forensic Science in Cyberspace*, P. Kanellis, E. Kiountouzis, N. Kolokotronis and D. Martakos (Eds.), IGI Global, Hershey, Pennsylvania, pp. 76–91, 2006.

[14] M. Rogers, J. Goldman, R. Mislan, T. Wedge and S. Debrota, Computer Forensics Field Triage Process Model, *Proceedings of the Conference on Digital Forensics, Security and Law*, pp. 27–40, 2006.

[15] P. Stephenson, Modeling of post-incident root cause analysis, *International Journal of Digital Evidence*, vol. 2(2), 2003.

[16] M. Suhanov, Linux for Computer Investigators: Pitfalls of Mounting Filesystems (`www.forensicfocus.com/linux-forensics-pitfalls-of-mounting-file-systems`), 2009.

[17] C. Tilbury, NTFS $I30 index attributes: Evidence of deleted and overwritten files, *SANS Digital Forensics and Incident Response* (`computer-forensics.sans.org/blog/2011/09/20/ntfs-i30-index-attributes-evidence-of-deleted-and-overwritten-files`), September 20, 2011.