SEAKER:

A Mobile Digital Forensic Triage Device

A Thesis Presented to

The Faculty of the Computer Science Department

California State University Channel Islands

In (Partial) Fulfillment

of the Requirements for the Degree

Masters of Science in Computer Science

by

Eric Elwood Gentry

Advisor: Michael Soltys

December 2018

APPROVED FOR MS IN COMPUTER SCIENCE

Advisor: Advisor Name **Date**

Name **Date**

Name **Date**

APPROVED FOR THE UNIVERITY

Name **Date**

Non-Exclusive Distribution License

In order for California State University Channel Islands (CSUCI) to reproduce, translate and distribute your submission worldwide through the CSUCI Institutional Repository, your agreement to the following terms is necessary. The author(s) retain any copyright currently on the item as well as the ability to submit the item to publishers or other repositories.

By signing and submitting this license, you (the author(s) or copyright owner) grants to CSUCI the nonexclusive right to reproduce, translate (as defined below), and/or distribute your submission (including the abstract) worldwide in print and electronic format and in any medium, including but not limited to audio or video.

You agree that CSUCI may, without changing the content, translate the submission to any medium or format for the purpose of preservation.

You also agree that CSUCI may keep more than one copy of this submission for purposes of security, backup and preservation.

You represent that the submission is your original work, and that you have the right to grant the rights contained in this license. You also represent that your submission does not, to the best of your knowledge, infringe upon anyone's copyright. You also represent and warrant that the submission contains no libelous or other unlawful matter and makes no improper invasion of the privacy of any other person.

If the submission contains material for which you do not hold copyright, you represent that you have obtained the unrestricted permission of the copyright owner to grant CSUCI the rights required by this license, and that such third party owned material is clearly identified and acknowledged within the text or content of the submission. You take full responsibility to obtain permission to use any material that is not your own. This permission must be granted to you before you sign this form.

IF THE SUBMISSION IS BASED UPON WORK THAT HAS BEEN SPONSORED OR SUPPORTED BY AN AGENCY OR ORGANIZATION OTHER THAN CSUCI, YOU REPRESENT THAT YOU HAVE FULFILLED ANY RIGHT OF REVIEW OR OTHER OBLIGATIONS REQUIRED BY SUCH CONTRACT OR AGREEMENT.

The CSUCI Institutional Repository will clearly identify your name(s) as the author(s) or owner(s) of the submission, and will not make any alteration, other than as allowed by this license, to your submission.

Title of Item

3 to 5 keywords or phrases to describe the item

Author(s) Name (Print)

Author(s) Signature Date

This is a permitted, modified version of the Non-exclusive Distribution
License from MIT Libraries and the University of Kansas.

SEAKER:

A Mobile Digital Forensic Triage Device

Eric Elwood Gentry

August 22, 2018

Keywords: Digital Forensics, Digital Forensics Triage, Mobile Digital Forensics, Digital Evidence, Digital Evidence, Forensic Tools, Raspberry Pi

Abstract

As our world of digital devices continues to expand, the potential for digital evidence available to law enforcement during case investigation is ever increasing. The growing amount of digital evidence, along with the deprived pool of Digital Forensic Investigators is causing a backlog to form at many of the digital forensics labs around the world. This backlog leads to delays in evidence analysis and reporting, causing investigators and prosecutors to postpone or even drop on-going cases.

The SEAKER device is a digital forensic triage tool that is designed to be simple, portable, inexpensive, robust, and easy to use. SEAKER is an acronym for Storage Evaluator And Knowledge Extraction Reader. Utilizing a Raspberry Pi, this is a novel approach to helping provide immediate feedback to investigators along with attempting to stem the backlog problem. It was originally developed for on-scene investigations that require immediate feedback, especially in time-sensitive investigations. It also appears to be an excellent tool to help reduce the backlog by preventing over-collection of digital evidence. SEAKER is not meant to replace a fully-functional digital forensic lab, but instead to augment the initial investigation and help reduce the backlog. This research and device overview proposes the mobile, inexpensive, digital triage device called SEAKER.

Contents

1 Introduction and Literature Review 1

1.1 Introduction 1

1.1.1 Author's Contributions 5

1.2 Literature Review 6

1.2.1 History of Digital Evidence 6

1.2.2 Process and Procedure Standardization 8

1.2.3 Reactive vs. Proactive Digital Forensic Investigation Processes 13

1.2.4 Aquisition Methodology 14

1.2.5 Analysis Methodology 15

1.2.6 Combined Aquisition and Analysis Methodologies 17

1.2.7 Digital Evidence Backlog 20

1.2.8 Digital Evidence Triage 23

1.2.9 How SEAKER Can Help 26

2 Background 28

2.1 Legal Details 29

2.2 Technical Details 31

3 Development of SEAKER Device 33

3.1 Conception 33

3.2 Setup Script For Raspberry Pi 36

3.2.1 Web Server 37

3.2.2 WIFI Setup 38

3.3 Rules For Mounting 39

3.4 Code for Searching Device 40

3.5 Web Code 41

3.6 Process Flow 44

3.7 Tools Used for Development 47

3.7.1 Hardware 47

3.7.2 Programming Languages and Scripting 48

3.7.3 Raspbian Operating System 49

3.7.4 Collaboration Tools 50

3.7.5 Setup Tools 51

4 Experimental Results 53

4.1 Prototype Demonstration 53

4.2 Results 54

5 Conclusions and Future Work 55

6 Appendix 67

6.1 SEAKER Setup 67

6.2 SEAKER Usage 74

6.3 Code 79

6.3.1 Directory and Filename Collection Code (in C) 79

6.4 Results of Testing 80

6.4.1 Collection Timing 80

7 STOP HERE - Abstract 84

8 Keywords and glossary 84

9 Graphs, Images, Figures, and tables 84

Glossary

Raspberry Pi A small, affordable computer created with the intention of providing low-cost computing power to the masses. , i, 5, 30, 31, 34, 36, 37, 46, 47, 48, 49, 51, 52, 67, 70, 71, 72, 73

SEAKER A digital forensics triage device built using a raspberry pi. It is an acronym for Storage Evaluator And Knowledge Extraction Reader. , i, ii, v, vi, 4, 5, 6, 14, 15, 16, 17, 26, 27, 32, 66, 68, 69, 70, 73, 74, 75, 77, 78

List of Figures

1 Integrated Digital Investigation Process Model (IDIP) 10

2 IOT Device Data Growth 20

3 Computer Forensic Field Triage Process Model (CFFTPM) 26

4 Main Page 42

5 Results Page 43

6 Default Keywords page 44

7 General Process Flow 45

8 SEAKER Creation Process 69

9 iPhone WIFI connection to the SEAKER. 74

10 Using a browser to connect to `http://seaker01.local` 76

11 The results of the search of a particular device. 77

12 The functionality of SEAKER from the user perspective. 78

List of Tables

1 SEAKER set up: Required Hardware and Software 68

2 Collection Algorithm Timing Data 80

1 Introduction and Literature Review

1.1 Introduction

Law enforcement investigations involve many aspects of criminality and need carefully thought-out procedures and practices. These procedures and practices are essential to finding the evidentiary information necessary to determine criminal liability, but are also in place to ensure that the evidence collected is not tainted and is sound, viable, admissible court evidence. Establishing and retaining the forensic integrity of the evidence is a required and crucial part of the investigator's task.

Performing investigations is also a noteworthy endeavor. There are many steps involved that require special training to be performed properly. One primary example is the *chain of custody.* This refers to the step-by-step documentation record regarding evidence that includes details such as who had custody of the evidence, when they had custody, who it was transferred to, who analyzed it, etc. Another is the exacting science of collecting, labeling, itemizing, and acquiring of evidence. For instance, collecting physical evidence requires the use of gloves, evidence bags, fingerprint-dusting equipment, etc. to prevent cross-contamination, fingerprint smudging, DNA evidence mishandling, and a multitude of other evidence tainting. Without the proper adherance to guidelines, even conclusive evidence may not be admissible during a case.

Digital evidence is also very essential to many investigations and cases in the modern world. With each passing year, more and more digital devices are collecting, storing, and uploading data. As well, electronic devices for personal use appropriately labelled the Internet of Things (IoT) or the Internet of Everything (IoE) are becoming more and more ubiquitous in our everyday lives. IOT devices are now everyday household items like refrigerators, thermostats, light bulbs, window coverings, garage door openers, keys, clothes, and much more. These devices and the massive amount of digital information that is being generated and collected are often helpful in criminal investigations. The data can be used to construct timeframes of activity, locations of individuals, Internet activity, computer users and usages, and lots other potential digital information.

One growing and particularly helpful aspect of an investigation is digital forensics. This not only involves collecting potential digital evidence, but also analyzing, and reporting procedures. This almost always requires a search warrant - a court-ordered search and seizure of potential evidence of a location where a suspect resides, works, or may be storing it. The search warrant is executed after it has been obtained from a judge and can involve physical and digital evidence, as well as other items of consequence.

Search warrant investigations are often fraught with danger, intentional obscurity, hidden evidence, and potential mishandling of evidence. Before anything else can be done, the location must be considered secure - considered safe from harming investigators and free of potential threats. Once a scene is secured at a search warrant service involving electronic evidence, three activities take place simultaneously: the search for physical evidence, the search of the physical evidence itself for electronic evidence, and the interviewing of involved parties.

The physical and digital evidence can guide the interviewing of the suspect(s), but also has the potential to have both positive and negative effects on the outcome of the investigation. If investigators do not locate any physical evidence for an examiner to evaluate, then intelligence is not gathered and the interviewer has less information with which to confront the suspect(s). If investigators present physical evidence to an examiner who is able to evaluate it quickly in the field, then the interviewer (who is oftentimes also the lead investigator on the case) can confront the suspects and potentially secure statements that lead to prosecution.

This leads to the need for digitial forensics specialists to bring their lab equipment into the field, especially when serving a search warrant. The lab equipment is specialized software and hardware designed to analyze, report, and maintain forensic integrity on potential digital evidence. This equipment often involved a laptop, a write-blocking device, media imaging storage devices, expensive software, and associated cabling for connection and power. As well, this software is designed for extensive and in-depth searching and often takes hours or days to analyze the evidence. Many of the reports from these systems are designed to be thorough and may take a skilled digital forensic examiner days to pour over the material produced. Oftentimes, this equipment is not brought into the field and all digital evidence is simply collected for later analysis at the law enforcement facilities.

The need for a more field-friendly digital forensic *triage* solution will assist in the initial investigation tasks in multiple ways:

1. It enforces a structured procedure and approach that is user-friendly to *non-digital forensic aware investigators* with the goal of simple instructions for use and very simple evidence location.

2. It enables investigators, especially interrogators, a very fast digital-evidence overview into the types of files and information being accessed and stored on the computer equipment at the site of the search warrant.

3. It limits the number of devices and therefore the amount of data required in the in-depth analysis phase at the lab.

4. It minimizes the impact and inconvenience to innocent parties at the site of the search warrant. The devices that are searched and found to have no evidenciary value can be deemed inconsequencial to the case and not be taken into custody.

5. It may be used to provide initial, albeit simplified, analysis results on potential digital evidence ingested into the digital forensic lab.

The topic explored in this paper is a portable, inexpensive, efficient device, named SEAKER, that is intended to overcome the need for a full digital forensic lab equipment suite to be brought into the field. The SEAKER device was conceived and an initial prototype was produced at the California State University Channel Islands (CSUCI) campus in a Masters level Cyber Security class (COMP 524, Summer 2017) in direct collaboration with the Southern California High Tech Task Force (SCHTTF) division of the Ventura Country District Attorney's (VCDA) office.

1.1.1 Author's Contributions

Author's direct contributions to the SEAKER device project:

1. Developed the bash script to turn a standard Raspberry Pi into a SEAKER device by programmatically installing raspbian software packages, setting up WIFI as a wireless access point, adding a web server, and preparing the running environment with the proper fileset

2. Wrote a custom executable using the C programming language to increase the SEAKER device's searching efficiency in lieu of slower, native operating system solutions for finding content on digital media

3. Co-presented and demonstrated the initial SEAKER prototype for the Summer 2017 Masters level CSUCI Security class project to SCHTTF, CSUCI department heads, and local community leaders

4. Presented the SEAKER device as a thesis project at the April 2018 CSUCI Cyber Security Event to CSUCI President Erika Beck, California State Assembly Member Jacqui Irwin, Ventura County Sherriff's Department, and other local community leaders

5. Co-authored a conference paper on the SEAKER project and the technology behind it

6. Updated and enhanced the SEAKER device functionality to support the

latest raspbian operating system (Stretch Lite, April 2018 release), including enabling ethernet passthrough to the wireless access point

7. Co-authored by creating Logic Models and assisting with read-throughs for a United States Department of Justice (DOJ) grant proposal for future work on this project (SEAKER) and a second, related security project (Voyager)

1.2 Literature Review

1.2.1 History of Digital Evidence

The digital forensics field began in the mid 1980s with an understanding from several law enforcement agencies that computers would play a critical role in future criminal investigations of the future. In 1993, the FBI hosted an international conference on computer evidence in Virginia. This was the first major conference on the subject and had attendees from 26 different countries. Much of the original computer forensics at that time related to recovering information from local computers.

Among the early pioneers in the digital forensics field, there was a common understanding that a system of processes and procedures were needed to locate, record, analyze and report information. This process would have to be similar to how non-digital physical evidence was handled, but also include other computer specific preservation methods to ensure the integrity of evidence found. Those processes and procedures have increased in complexity over time and have suffered from the lack of unanimous adoption to a single standard.

In 2006, Rogers et al[9] proposed a standardization model for a portion of the entire process called *triage* for digital forensic examiners to follow: Computer Forensics Field Triage Process Model. The authors of the CFFTPM note the important legal and technical considerations prior to implementing CFFTPM on a particular investigation. The legal considerations include issues related to search warrant scope and its limitations, U.S. Constitutional 4th Amendment rights, etc. The technical considerations include type of case, criticality of timeliness, skillset of the on-site digital forensic examiner, skillset of the suspect, having proper lab equipment on-site, scene control, etc.

Even as late as 2013, Shaw et al[10] points out that neither digital forensic triage examination nor digital forensic full examination are well defined. Digital forensic triage may mean something completely different to two digital forensic examiners. As well, full digital forensic examination has no robust standard to follow, Although there has been no shortage of attempts.

The National Institute of Standards and Technologies (NIST) published guidelines for several different types of digital evidence, for instance mobile phones, and computers. However, NIST focuses on the analysis portion of the science, but leaves the collection, and reporting aspects unexplored. The International Standards Organization also published a set of guidelines, but primarily focused on collection and handling aspects of digital evidence. Ajijola et al[1] provided a thorough review in 2014 of the NIST SP 800-101 Rev. 1:2014 guidelines titled

Guidelines on Mobile Devices Forensics and ISO/IEC 27037 titled Guidelines for Identification, Collection, Acquisition, and Preservation of Digital Evidence. Their recommendation was a combined approach, though still not a fully formed solution.

1.2.2 Process and Procedure Standardization

Several approaches over the years have been proposed as universal processes and procedures to gathering, reviewing, and presenting digital evidence. These approaches range in number of steps, process coverage, and overall methodology, but all have the common goal of finding usable digital evidence for preventing future harm to society and preserving the potential digital evidence's integrity for means of presentation in court cases.

In the early years, some research facilities arose to help create and define the processes and procedures necessary. Among these were the Computer Analysis and Response Team (CART), the Scientific Working Group on Digital Evidence (SWGDE), the Technical Working Group on Digital Evidence (TWGDE), and the National Institute of Justice (NIJ). Since their respective inceptions, they have all strived and contributed to standardization on approaches and methods for the handling and processing of all digital evidence.[7].

In addition, the United States Department of Justice (DOJ) published *Electronic Crime Scene Investigation: A Guide to First Responders* [2] in the early 2000s that outlines the necessary four steps for properly investigating digital Evidence:

1. *Collection*, which involves searching for digital evidence, deciphering what should be collected, acquiring the media, and chain of custody documentation.

2. *Examination*, which includes searching the digital media and attempting to reveal the evidence, especially when it is hidden or obscured.

3. *Analysis*, intending to review the evidence for important legal infringements.

4. *Reporting*, for documenting the process used and evidence uncovered in the investigation.

The 2003 *Integrated Digital Investigation Process* (IDIP)[4], proposed by Carrier, et al is a model that in their own words:

> "uses the theory that a computer is itself a crime scene, called the digital crime scene, and applies crime scene investigation techniques."

It consists of five main phases (in addition, see Figure 1):

1. *Readiness*, for training, preparedness, infrastructure and resources preparation before any investigation even begins.

2. *Deployment*, which is intended to capture the process for when an incident requires digital evidence procedures and assignment of resources.

3. *Physical Crime Scene Investigation*, for processing the physical evidence from the environment.

4. *Digital Crime Scene Investigation*, for collecting and analyzing the digital evidence that exists in the virtual environment.

5. *Review*, for examining the process used in the investigation and potential areas for improvements.

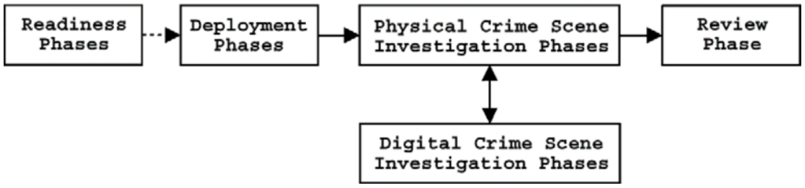

Readiness Phases
Deployment Phases
Physical Crime Scene Investigation Phases
Review Phase
Digital Crime Scene Investigation Phases

Figure 1: Integrated Digital Investigation Process Model (IDIP)

Baryamureeba and Tushabe[3] proposed a model based on the best parts of the US DOJ's *Electronic Crime Scene Investigation*, the *Abstract Digital Forensics Model*, and the IDIP model. Their proposal was called the Enhanced Integrated Digitial Investigation Process (EIDIP). Their model was proposed in 2004 at the annual Digital Forensic Research conference (DFRWS)[3] and included the following phases:

1. *Readiness*, same as IDIP.

2. *Deployment*, which encompasses the Deployment, Physical Crime Scene Investigation, and Digital Crime Scene Investigation phases from IDIP.

3. *Traceback*, that includes connecting the evidence collected in the previous phase to the suspect(s). Typically this is done with IP addresses and requires the authority to gather this information.

4. *Dynamite*, which includes the reconstruction of the events suggested by the evidence and the documentation and submission to the appropriate legal authories.

5. *Review*, same as IDIP.

In 2006, Rogers et al[9] proposed a reliable, repeatable process model designed specifically for digital evidence triage called Computer Forensics Field Triage Process Model (CFFTPM). It was created in partnership with Purdue University's Cyber Forensics and Computer and Information Technology Departments, along with the National White Collar Crime Center[9]. The process was derived from several other military and law enforcement models including IDIP, Digital Crime Scene Analysis (DCSA), and a military Operations Order (OpOrd). In coordination with the Southern Indiana Assistant U.S. Attorney's office, USADA Steve Debrota, Rogers et al[9] implemented and reported on the success of their proposal. This amalgamation of approaches is still inspiring the latest trends in Digital Forensic approaches.

Shaw et al[10] analyzed the ACPO and focused on the second step (Capture) as the primary guideline for evidential integrity. They strongly suggest compliance with digital forensics best practices, like the ones provided in the ACPO. Their final approach and recommendation was to combine Linux utilities with a simplistic interface to standardize the output and enable investigators who may not have full digitial forensic backgrounds to perform triage on potential digital evidence.

The Association of Chief Police Officers (ACPO), a private company that helped establish and develop policing practices in England, Wales, and Northern Ireland for many years, put together a *Good Practice Guide for Digital Evidence*[11] in 2012 that outlines some recommended procedures for dealing with digital evidence. As with other methodologies, this guide explains utilizing a four step approach: Plan, Capture, Analyze, Present.

The ISO/IEC 27037 guidelines provide an attempt at an internationally recognized approach, with the goal of making it easier to compare, combine, and contrast results for out-of-jurisdiction cases and for data scientists' research. It provides a common reference line for digital forensics[1]. However, it is not meant to replace laws or regulations. The main purpose is to provide practical assistance for investigations involving potential digital evidence, while preventing digital evidence corruption. This process facilitates the usability of evidence by other jurisdictions. This guideline provided four steps for handling potential digital evidence: Identification, Collection, Acquisition, and Preservation. However, this is incomplete, as it only addresses gathering, not actually evaluating or providing results to law enforcement investigators.

All of these approaches are intended to help find and gather digital evidence. They are also intended to maintain the forensic integrity. Some organizations combine these approaches into a system of processes and procedures intended for use in their own facilities. Unfortunately, many organizations have home-grown solutions passed down from senior members of the digital forensics team to the newer team members. Not having a universally recognized and accepted standard leads to complications and difficulties when digital evidence needs to be shared across other jurisdictions and boundaries[1].

1.2.3 Reactive vs. Proactive Digital Forensic Investigation Processes

Reactive digital forensic investigation processes are utilized after an offense has been committed to help identify the charges and suspects. This is the most common process for digital forensics. The *proactive* digital forensic investigation processes are to attempt to detect before or during an active offense is committed. This is not a job for a typical law enforcement investigator.

This SEAKER research is based on the reactive digital forensic investigation process in the hopes of reducing the digital forensic lab backlogs across the country and world in two ways. The first way is to reduce the amount of digital evidence acquired for the digital forensics lab by enabling efficient and effective on-site triage to occur by utitizing the commonality of digital evidence collection and analysis into a single step. The SEAKER digital evidence triage tool enables an initial collection of information and subsequent searches by any number of local, on-scene investigators. These investigators do not need extensive training in digital forensics to utilize it.

The second way is to help reduce the existing backlog by enabling a faster, more streamlined approach to initial potential evidence gathering and reporting.

This approach utilizes the SEAKER digital evidence trial tool to perform an initial acquisition and analysis on every exiting case to provide a “first-look” at the information. This also enables a searching mechanism within a few minutes of plugging in digital evidence media to enable digital forensic investigators a quick review of materials. The process will help with prioritization of evidence, a basic analysis and potentially initial evidence in the form of a report that can be provided to investigators and prosecutors.

1.2.4 Aquisition Methodology

The SEAKER tool is designed specifically for the *triage* phase of digital forensic investigations. As a general approach, this research splits the act of dealing with digital evidence into two separate methodologies: *aquisition* and *analysis*. These are the main foci since the SEAKER device is designed to do both in a very timely fashion, with comprehensive searching results.

Aquisition has multiple definitions across the digital forensic universe. Some define this as the process of gathering the digital devices, logging them using the *chain of custody* paperwork, and removing them from the location denoted in the search warrant. Others define this as the collection of potential evidence from the digital devices. However, in this case it is meant to imply the entire set of processes and procedures from training the digital forensic examiners and SEAKER users to collecting the media that potential digital evidence may be stored on, to the gathering of potential evidence from each digital device.

The set of processes and procedures for training and usage of the SEAKER device is well documented in the appendix, but could also be supplemented by more specific guidelines that combine the digital forensic lab's procedures with the usage documents.

When a search warrant is being executed, the SEAKER device is intended to be used to help manage the triage process of the potential digital evidence. This helps on-scene investigators determine priority, worthiness of custody, and documentation of the digital media[5].

The *collection* phase of aquisition specifically highlighted here is the gathering of physical digital media information and the capturing of potential digital evidence content in the triage environment. The setup and training materials for creating and using the SEAKER device are referenced in the appendix.

1.2.5 Analysis Methodology

The *Analysis* methodology is considered the second phase in the SEAKER approach and can consist of both the *triage* analysis stage and the *full* analysis stage. Specifically, this paper and the SEAKER device focus on the triage stage of Analysis. This stage can be applied in the field or at the digital forensics lab, while the full analysis is unlikely to be accomplished in the field. A full analysis could take many hours or days and is not the first priority when serving a search warrant.

Digital evidence triage analysis is a very useful part of an investigation and can be implemented using the SEAKER device. The triage analysis stage is considered in this research to consist of an interactive web page that detectives and investigators can use to lookup search terms from the digital evidence collected from suspect devices plugged into the SEAKER device. It also consists of the reports that are generated and stored on the SEAKER device. The reports consist mainly of the metadata about the digital media and the collection statistics, but does not include specific digital media content.

Some previous versions of analysis included specific types of operating systems. Rogers' et al[9] research in 2006 covered the primary machine type at the time: the standard Windows machine. Unfortunately, this leads to an outdated model over time, since the processes and procedures become obsolete as new technology arises. Along with the proliferation of IOT devices, new technologies also have emerged as more mainstream that need to be incorporated into a more generalized approach. More operating systems are being utilized on a regular basis, like Linux, unix versions, and Apple OS. In fact, even our SEAKER device is an IOT device based on a unix variant of Debian.

Other versions of analysis research deal directly with specific types of devices. Ajijola et al[1] reviewed the NIST guidelines that provide an in-depth look into mobile devices, helping to explain the technology involved and its relationship to the forensic process. NIST itself is a technological, non-regulatory federal agency under the U.S. Department of Commerce. The NIST process model labeled NIST SP 800-101 lays out the digital evidence procedures in four steps: Preservation, Acquisition, Examination and Analysis, and Reporting[1]. These four steps provide the necessary steps for the digital evidence process model as a suggested way to evaluate mobile device information. This is useful, but not complete for law enforcement investigators.

The SEAKER device achieves the goal of being device and operating system agnostic in terms of being able to collect content information from digital media. It currently supports multiple operating systems, as well as multiple device types (provided there is a way to adapt the device to USB). The only other limitation to the SEAKER device's ability to read the content is the availablity of a device driver that allows mounting on Raspbian lite's operating system.

1.2.6 Combined Aquisition and Analysis Methodologies

TODO: finish this section
Hitchcock et al[5] has proposed and evaluated a "tiered forensic methodology" model that defines a process of digital forensic triage utilizing non-digital evidence specialists. In their research, they identified a large and growing backlog of digital evidence. This backlog has led to problems in the law enforcement community with regards to collecting, analyzing, reporting, and prosecuting.

The next tier is when the already-triaged digital evidence is sent for full evaluation. This is a certified facility that can perform full digital forensic analysis, called a Technological Crime Unit (TCU). The TCU is currently heavily inundated with cases needing analysis and reporting of digital evidence.

This tiered approach is based on a Computer Forensic Field Triage Process Model proposed by Rogers et al [9] and the internation standard ISO 27037 (Information Technology - Security Techniques - Guidelines for identification, collection, aquisition, and presentation of digital evidence). The process model breaks down the six phases of digital evidence categorization, which Hitchcock et al[5] loosely based their four phase approach on. The four phases are: planning, assessment, reporting, and threshold. The ISO 27037 standard specifically attempts to address the need to minimize the risk of potential digital evidence being spoiled by mishandling, while also attempting to maximize the evidentiary value of digital evidence collection.

This approach is not without risks. One concern is the accidental exclusion of an item of digital evidence that is important to the investigation. Another is the level of computer skills and training of the DTF expert. The paper ??? does attempt to mitigate the latter with training and management process, while providing evidence that the former is a common misconception in most cases.

Shaw - One example is to note encrypted compressed files for review later.

Rogers - This process in no way supersedes the ability or need to perform a full forensic examination at a full-featured digital forensic lab.

Ajijola et al, 2014[1] also proposed a new process model that is a hybrid of both models with the resulting combination being much more effective than either of its individual parts.

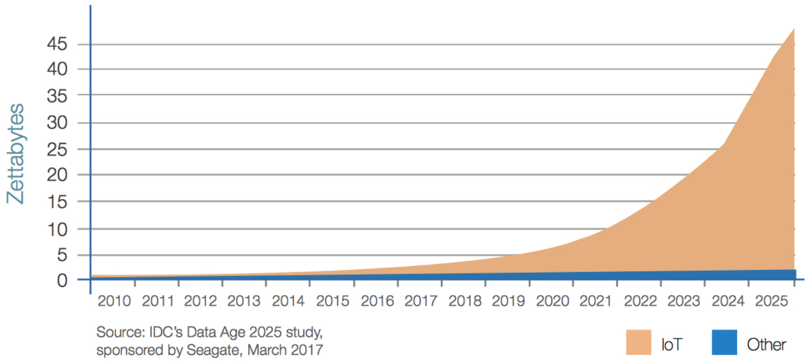
In the research for combining the NIST and ISO guidelines, Ajijola et al[1] explores the commonality, differences, and limitations of each model. Although both models follow the Auditability, Repeatability, Reproducability, and Justifiability requirements, as well as the Confidentiality, Integrity, and Availability standards, they individually lack some necessary phases to enable them to be used separately. The NIST process model lacks the Identification and Collection phases, while the ISO process model lacks Examination, Analysis, and Reporting aspects of a full Digital Evidence processing model.

The combination of these two approaches, as suggested by Ajijola et al[1], provides a new five step approach: Identification, Collection and Acquisition, Preservation, Examination and Analysis, and Reporting. These steps provide a more comprehensive approach that law enforcement can use to fulfill its evidenciary duties in an investigation. When both process methods are used, the goals approach a full set of tasks from initial on-scene evalutation to the end of the in-lab digital forensics investigation.

1.2.7 Digital Evidence Backlog

TODO: finish this section

IDC forecasts that by 2025 the global datasphere will grow to 163ZB (see Figure 1) (that is a trillion gigabytes). That's ten times the 16.1ZB of data generated in 2016. All this data will unlock unique user experiences and a new world of business opportunities.



Source: IDC's Data Age 2025 study, sponsored by Seagate, March 2017

Figure 2: IOT Device Data Growth

Some of the current challenges in digital forensic investigations are directly related to the amount of data being created. As Lillis et al[6] explores in their research, there are three main factors involved in the digital forensic backlog: increasing number of devices seized per case, increased number of cases involving digitial evidence, and the increasing volume of data per digital media. This has lead to a growing and already substantial backlog in digital forensic investigations.

One effect of this increased delay and backlog is that cases become inactive, waiting for new leads. A more aggressive approach to solving the backlog could help prevent dismissals, cold cases, and potentially more societal harm from a corrupt investigation suspect.

Raghavan[8] has accumulated a list of 5 major challenges that the digital forensics community is facing and continue to add to the backlog problem.

The first is the complexity of binary data aquisition, i.e. low level data aquisition through digital media duplication. This challenge causes the need for sophisticated data reduction techniques.

Another complexity is the diversity of data and lack of standard examination techniques. The plethora of operating systems and file formats has been increasing and is posing a more and more significant challenge over time.

The consistency and correlation problem is yet another challenge. This is a problem resulting from the current digital media investigation tools not providing the entire picture to investigators. Only part of the whole picture is provided when these tools find digital evidence.

Another issue that Raghavan[8] proposed is the volume of data to sort through. The sheer amount of data that exists per user is increasing at an alarming rate [cite?], and has lead to a very large backlog of digital evidence to investigate. These delays have even caused some cases to be dismissed. This challenge is exacerbated by the lack of adequate automation for digesting the data.

The fifth, but certainly not the last, challenge proposed by Raghavan[8] is the timeline synchronization issue with digital evidence. Since the evidence could be collected in different time zones, with different timestamp formats, clock skew, etc, lining up the events in order can be challenging or infeasible.

With the proliferation of Internet Of Things (IOT) devices and cloud storage, the field of digital forensics continues to expand. These areas pose a great challenge, but also new opportunities. Lillis et al[6] researched cloud storage and found some areas of opportunity, for instance parallel processing, distibuted computing, GPU/FPGA utilization, and others. These areas for increasing the efficiency of ditigal forensics can be explored further due to the substantially reduced I/O limitations in cloud storage.

The Internet of Things (IOT) also poses new challenges. IOT devices are estimated to number near 40 billion by 2020, contributing to the overwhelming amount of digital data. Since these devices tend to have more non-persistent memory and less storage, this causes added complexity for gathering and analysis. In addition, a portion of IOT devices are battery operated and computationally challenged, leading to loss of data over time.

Hitchcock et al[5] - The backlog and delays in case reporting are contributors to a common problem of time sensitivity. Some countries have given their citizens a right to a “speedy” trial. As well, some countries have statutes of limitation (limits on how long after the crime was committed to resolve the case) for most crimes. Some administrative situations are also contributors, for instance case prioritization based on chronological filing, crime severity, or victim needs.

1.2.8 Digital Evidence Triage

TODO: finish this section

The summary of the research done by Hitchcock et al[5] are as follows. They sought to expedite the process of sending digital evidence for analysis and results. One of their goals is to enable more field triage of digital evidence to reduce the amount collected, and act specifically on pertinent information only. They recommended that some front-line crime scene investigators (non-forensic analysts) be trained in the implementation of digital evidence triage and evaluation. These trained individuals would be Digital Field Triage (DTF) experts and have the ability perform field-level digital evidence triage. This triage would specifically weed out the benign from the consequential digital evidence with high certainty, while also protecting the digital evidence from spoilage and preserving evidentiary integrity.

One digital evidence triage method proposed by Shaw et al[10] seeks to standardize on an approach they call “enhanced previewing”. Enhanced previewing seeks to solve some of the problems associated with typical triage approaches. As is the case in other research, Shaw et al[10] extolls the need to reduce digital forensic evidence analysis backlogs, especially with the evolution of big data and the proliferation of digital devices.

The proposal for a practical and robust methodology by Shaw et al[10] aims to stem the concerns of a typical triage process. Risks still exist, for instance overlooking digital evidence, but it is argued that those risks are outweighed by the risks of a lengthy process due to large backlogs and the associated delays in evaluating that evidence. Another concern exists that inadequately trained people will be charged with performing on-site digital evidence triage and mishandling or incorrectly evaluating results will cause evidence spoilage. Other concerns are the potential high cost of software and training.

In order to provide a simple, yet robust mechanism, Shaw et al[10] starts with an open source, CD-bootable image of GNU/Linux and enhances its features to include boot-time application launching, and a simple to use interface with minimal ability to deviate from task. This bootable CD is intended to be placed into evidenciary computer systems and booted using a series of BIOS modifications or boot-time interruptions. This mechanism to boot the system off of a bootable CD is difficult, and where the most problems with untrained users of the enhanced previewing will happen.

The enhanced previewing concept has valuable merit, in that the collection mechanisms are thorough. Using the GNU/Linux based system and having written code for it, Shaw et al[10] utilized some well thought-out approaches. First, all hard drives from the evidentiary system are mounted into the GNU/Linux filesystem as read-only, thereby eliminating the need for write-blockers. As well, the entire hard drive is evaluated, including the file system, all partitions, unallocated space, deleted files, and compressed files. In addition, other mechanisms are employed that continue to enhance the previewing are employed.

Rogers - The CFFTPM was created to enhance the investigators ability to obtain useful information at execution time of a warrant at the suspect's dwelling or work. The process is designed to be used in the first few hours of the investigation, especially during the first suspect interview and search execution phase of the investigation. It is known that suspects are more likely to divulge more information and be more cooperative in that environment (Yeschke 2003[12]). As well, location of and presentation with suspect "triggers" from the potential evidence increase the suspect's willingness to talk and cooperate while on site.[9]

The foci of the CFFTPM are immediately finding usable evidence, identifying victims at acute risk, guiding the on-going investigation, identify potential charges, and accurately assess the offender's danger to society.

Rogers - The CFFTPM is broken up into phases, two of which have subphases (see Figure 2). The main phases consist of Planning, Triage, Usage/User Profiles, Chronology/Timeline, Internet Activity, and Case-Specific Evidence. Usage/User Profiles are broken down into three sub-phases: Home Directory, File Properties, and Registry. These are important and help distinguish user specific activity and permissions. The Internet Activity phase is also broken down into three sub-phases: Browser Artifacts, Email Artifacts, and Instant Messenger Artifacts. These also help establish user activity. Some importance is explicitly stated to skip based on type of investigation and prioritizing the investigation.

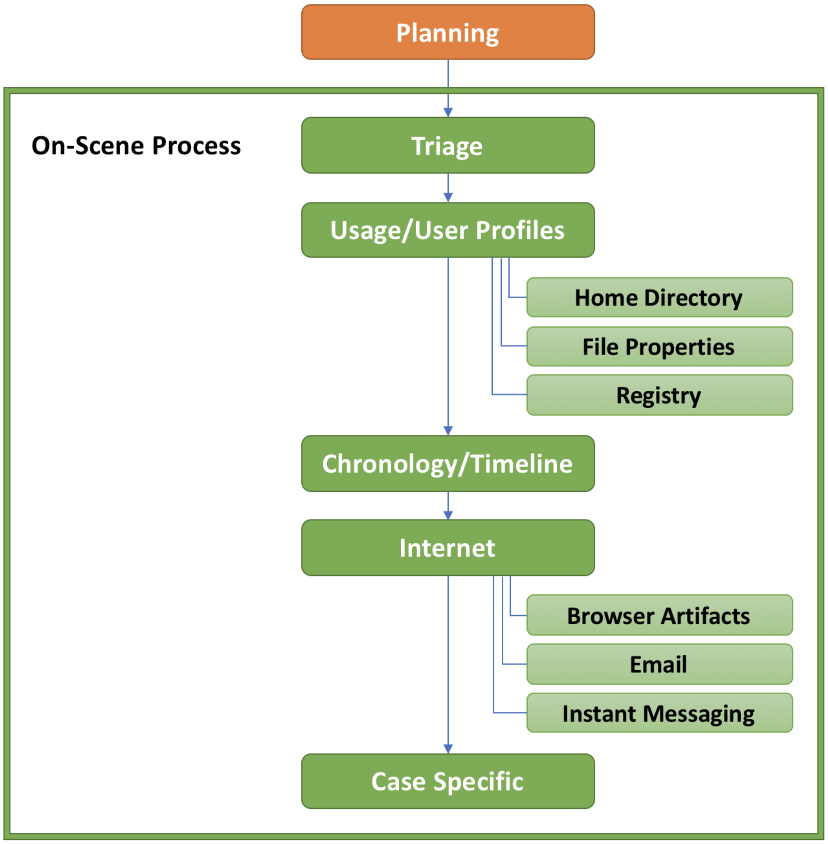


Figure 3: Computer Forensic Field Triage Process Model (CFFTPM)

1.2.9 How SEAKER Can Help

TODO: finish this section

The research of Hitchcock et al[5] should be referenced for a good process starting point for digital forensic labs.

Rogers - This is where the SEAKER portable triage device can help by evaluating every aspect and prevent the on-site investigators from skipping or deprioritizing critical potential evidence.

Rogers - This is also an area where the SEAKER portable triage device can help eliminate some of the potential problems, for instance technical prowess of the on-site investigators and proper lab equipment on-site.

Rogers (CFFTPM) - Some particular aspects of the phases are critical to investigators in revealing evidence or potential evidence for the SEAKER portable traige device. Usage/User Profile information is extremely important. This includes the need to be able to view and search files, folders, registry keys, and file properties associated with a particular user. The Internet Activity artifacts also become very useful, especially in the case of child pornography. The browser, email, and Instant Messaging artifacts can lead directly to potential charges. Finally, a Chronology/Timeline understanding and ability to sort based on it can significantly narrow down the possibilities of which user information and which Internet Activity is the most important and critical to the investigation.

Rogers (CFFTPM) -Implementing this in the SEAKER portable triage device is crucial for simplicity and ease of use. As well, it goes a long way towards having an implementation of SEAKER being understood and adopted. Reporting is also a critical need and is implemented in a way that will enable digital forensic investigators to provide early information to investigators and prosecutors. This helps alleviate the need to wait until the backlog of digital evidence is cleared to get any information from case-specific digital evidence.

2 Background

In a labratory environment, digital forensics investigators have the ability to discover a tremendous amount of material that is potential evidence. This includes anything digitally stored on the evidentiary media from explicity illegal files to IP address connections to a digital chronology of events.[8][9] The software and hardware necessary to perform the in-depth, full evaluation of the media are specialized for digital forensics work, but typically are costly, don't travel well, and can take many hours for results from a single digital media device.

In the field environment, digital forensics investigators are likely to not have the time, equipment, or proper environment to obtain evidence from the digital devices found during the execution of a search warrant. In some cases, due to various reasons, digital forensics investigators are not able to attend and therefore all of the digital media is taken into custody for analysis at the digital forensics lab. When they are able to attend, they typically bring a subset of their lab environment with them to start deciphering the digital information and attempt to perform a traige analysis.

In a coordinated effort with SCHTTF, this research is an attempt to help solve some of the field environment limitations of digital forensics investigators.

2.1 Legal Details

In order for digital forensic investigators to obtain the data from a digital device, a search warrant for that device must be obtained. As well, law enforcement must obtain a search warrant to acquire the device for searching in the first place.

Search warrants are necessary for law enforcement to obtain crucial evidence in many cases. A search warrant is obtained by a judge's order. Law enforcement must provide probable cause that a crime was committed and that items connected to that crime are likely to be found in the place specified by the warrant. The judge will review the matter and if they are in agreement, will authorize law enforcement to search a particular location for specific items which are declared in the search warrant.

When law enforcement aquires the digital devices for an investigation, there is an important document that must also accompany them. This is called the chain of evidence or chain of custody, which is used to track who currently has control of the evidence and who has had control chronologically since it was originally seized. The information in the chain of evidence is date, time, and location of acquiring, securing authority, and who is gaining posession of the device(s).

After a digital device has been acquired by digital forensic investigators, they must take special care not to alter the device's information in any way. This requirement figuratively mimics the care a physical forensic investigator must take to preserve physical evidence. Special processes must be followed to ensure that the potential digital evidence is not altered and, in fact, must be able to be proven if the matter ends up in a trial. This is critical to ensure that the evidence obtained from the device is admissible in court.

One device to aid in the proper handling of digital evidence is called a write-blocker. Digital forensic investigators use this device as an intermediary between the devices and the computer systems they plug the devices into for investigation. The typical first step when a device is acquired by a digital forensic lab is for the device to imaged (or copied bit by bit) so that the image can be used for further evidence searching. This provides an extra level of abstraction, so that the actual device is kept in pristine digital condition.

Write-blocking has been implemented in digital forensics labs with an in-line piece of hardware. Companies like Guidance Software and others have created write-blocking devices that are added to the list of hardware necessary to prevent modification of any kind to the potential digital evidence. The Tableau product line is a great set of these types of devices. Although they are made to be simple and easy to use, they create yet another piece of the lab that must be carried into the field and required to be plugged in and used.

For this research, the SEAKER device is intended to be used for triage investigation on digital devices to perfom an initial search for potential digital evidence. Instead of having a separate write-blocking device, the Raspberry Pi is setup with write-blocking capabilities when digital devices are connected to it. This configuration enables the SEAKER's own system to act as a software-write-blocker to prevent any digital alteration to the device.

2.2 Technical Details

Choosing the Raspberry Pi as the base platform for the SEAKER device was intentional and guided by some of the principals of the originating company.

Raspberry Pi is manufactured by the Raspberry Pi Foundation in the United Kingdom for the purpose of teaching Computer Science in schools and around the world. It is a fully functional CPU with RAM, input and output connections, status lights and the ability to be powered by batteries, USB, or an electrical wall socket connection. The form factor is small and the cost is kept to a minimum for ease of aquisition, use, and adaptability. As of this writing, the most powerful version of the Raspberry Pi is $35. Uses for the Raspberry Pi device are numerous and growing.

The SEAKER device project is yet another adaptation of how the Raspberry Pi device can be utilized from concept to fully featured digital evidence triage device.

Another goal of the SEAKER project is to enable investigators without digitial evidence training or with limited computer training to utilize it on-site at the execution of a search warrant. The SEAKER device is designed to be self-sufficient and automatically self-preparing when it is plugged into a power source. The device will boot, prepare the web server, the wifi hotspot and be enabled to handle digital devices that are attached to its USB port. Once a digital device is plugged in, it is automatically mounted and scanned. A web-page interface was created for accessing the scanned devices when a portable wifi-enabled phone or tablet are connected to SEAKER.

These attempts to make the process as simple as possible are intentional and make the process of digital evidence triage collection and searching accessible to investigators with or without specialized computer knowledge or training.

Some specialized training is available to those investigators who want to become digital forensic investigators. The typical learning takes place over a full year of classes and hands-on work through one of several federal or law enforcement agencies. These lead to certifications like Digital Evidence First Responder (DEFR), Digital Evidence Specialist (DES), or Digital Forensic Investigator (DFI).

3 Development of SEAKER Device

The SEAKER device concept is very novel, not only in its capacity as a digital forensics evidence triage device, but also in the fact that it is low cost, highly available, simple to setup and use, and provides very fast results.

There are other digital triage tools on the market, but almost every one is a software solution that involves either a separate laptop or a bootable CD, DVD or USB drive to enable the interaction. These types of software tools typically require advanced computer knowledge and a digital forensics specialists to be involved.

Since the SEAKER device project was a collaboration with SCHTTF, it already has built-in law enforcement acumen related to digital forensics. As a digital forensics evidence traige device, it could be shaping the way investigators handle computers and other digital equipment during execution of a search warrant.

3.1 Conception

The SEAKER device project came to be in the Masters level Cyber Security class (COMP 524) at CSUCI in the Summer semester of 2017. It was proposed to the class by professor Dr. Michael Soltys during one of the initial lectures as the final project for the course. The attending students agreed and work began on it, in addition to the other assignments due in the course. Dr. Soltys broke down the problem into categories so that teams could form and work on each piece individually. The categories were:

- Connecting hd to a RP/NUC, sensing OS and mounting (2 teams together)
- Searching in the mounted file system (2 teams together)
- Sending report to an iPad/laptop/handheld
- Documentation and troubleshooting
- Testing

This breakdown helped guide each team to get started on their contribution to the final project. However, before the students could begin, the device platform to use, the technological method for input and output, and the feature set had to be agreed on.

The device platform chosen was the Raspberry Pi. This enabled the students to work on the platform independently, due to the inexpensive nature of it. As well, two Raspberry Pis were provided for the classroom by SCHTTF and Dr. Soltys, respectively.

The input method chosen was setup for two types of input. The first type was connecting the digital devices to the SEAKER device. This was agreed upon to be either with the USB port that was built into the Raspberry Pi or via a USB to SATA converter cable. This enabled the digital device to be mounted by the Raspbian Operating System and automatically searched for content via the mounting rules. The second type of input was human input for a set of terms to search. The search terms were agreed to be put into a web form that would be submitted to the on-board web-server.

The output method clearly needed to match the input method in terms of technology, so the use of the on-board web-server was chosen to be the output method. When investigators are using the SEAKER device, they would be shown a webpage asking them to submit a set of search criteria. The results would be given back to the phone or tablet in HTML. This also enabled quick building of the HTML framework and response mechanisms.

Finally, the feature set needed to be agreed to. With direct guidance from the SCHTTF, specifically Frank Lyu, the class agreed to the following:

- Write-blocking of attached digital evidence devices
- SATA and USB storage devices
- FAT, NTFS and EXT* file systems to be read from storage devices
- Filename-keyword filter
  - Prepopulated keyword list
  - Customization of keyword list
- Status lights for power, and device status
- Wireless connection to a phone or tablet for keyword input and results
- Ability to find and display search results and digital device hardware information

The proposal for the SEAKER device project initially came from the SCHTTF. They wanted a device that could quickly ascertain potential evidence and enable on-scene investigators to search devices while questioning suspects. This process of searching the digital devices immediately is called triage. With it, investigators are able to provide actionable intelligence quickly, prioritize devices to be previewed, reduce preview setup time, and triage larger amounts of devices.

3.2 Setup Script For Raspberry Pi

The idea of a setup script for the Raspberry Pi was conceived early by the author, since each team was assigned to work independently and the deadline of implementation was extremely short. This was, after all, a summer class. In order to get everyone in the class up and running and able to do work on their individual pieces, there needed to be a baseline for everyone to start working with. Starting with the base operating system image, called Raspbian, the setup script was meant to modify it to handle the scenarios we were attempting to create.

First, there needed to be some initial setup for keyboard, timezone, SSH, hostname, and installing some additional packages. To prevent unnecessary software from occupying the local Micro SD card, the “lite” version of Raspbian was chosen as the base operating system. However, that meant that additional Raspbian packages needed to be added; for instance, the Apache web server, a DHCP server, PHP, the software to convert the wireless NIC card to an access point, and the device drivers for FAT32, NTFS, HFS, EXT*, etc.

Next, the setup script needed to customize the SEAKER device based on user parameters. These include hostname, IP address, DHCP supported range, Raspbian user *pi*'s password, and the wireless access point password.

The setup script then prepares the Raspberry Pi access point configuration, web server configuration, mounting rules, default web-pages, and compiles the custom C code for searching (listed in appendix).

Finally, in order to avoid simple hacking and password locations, the setup script clears the history, sets itself up to be deleted at boot time, and removes any other reminants from the original setup.

Most of this was done by the author and published to the class so that they could begin using the Raspberry Pi as a SEAKER device and begin their work to implement the rest of the functionality.

3.2.1 Web Server

The Apache web server was chosen, since it is a standard Unix-based operating system choice for serving web pages. It also supports backend coding opportunities when coupled with a server-side code execution program like PHP. Setting this up was included as a part of the setup script.

A couple of steps were needed to implement the web server. The first was to load the Apache and PHP packages coinciding with the Raspbian operating system. Since it is a branch off of Debian operating system, these were easily found and worked well. The next step was to load all of the files that were needed for the HTML and PHP to display and operate properly. The final step was to modify the access to each of the files to be specifically accessible by the web server daemon account. This was necessary to ensure the files could be read and served up by the web server when requested.

In addition, the *collection* code for searching the drive needed to have access to a shared location for the filename and folder searching algorithm. The \tmp folder was chosen as a suitable location, since both the collection program and the web server have access to it. An extra feature of using \tmp is that the operating systesm clears out the entire folder everytime it boots up, causing the previous data to no longer show up.

3.2.2 WIFI Setup

The wireless NIC also needed to be setup to be a wireless access point so that investigators could connect with a phone or tablet and use the web page access to perform searches on the digital devices. The main idea here was to have a password-protected closed network where the potential evidence could be searched. This was included as a part of the setup script.

The steps involved here were complicated and difficult to setup properly. As with the web server, the proper Raspbian operating system packages needed to be acquired and installed. In addition, the setup of those packages required setting up DHCP, WPA, the wireless NIC, and the access point daemon. Setting these up mainly required adding and altering text configuration files. Unfortunately, the Internet was not much help, leading to a lot of trial and error testing to make sure it all worked.

Finally, this process required a reboot, which was able to be postponed until the end of the setup script.

3.3 Rules For Mounting

During the setup script for the SEAKER device, auto-mounting is setup to automatically mount new digital media devices that are plugged into the USB port. Another script was written to handle the post-mounting *collection* of the applicable drive contents. The post-mounting script is also configured to run once any new digital media devices are plugged in.

In order to accommodate the request to ensure the forensic integrity of the suspected digital evidence, special mounting options were required. There are two different aspects for how a drive can be written to. The first is the standard *writable* option, which allows the content of files and folders to be created and modified. The second is called *journaling*, which is an operating system concept for logging when and what the OS did to the drive. Both options, *ro* and *noload*, are applied to the mounting options. This makes the SEAKER device functionally consistent with a write-blocking device, as mentioned earlier in Chapter 2.

3.4 Code for Searching Device

The code for searching the digital devices was specifically aimed at gathering every filename and location into a searchable file and storing it so that those drive contents could be searched, even after the digital device had been disconnected from the SEAKER device.

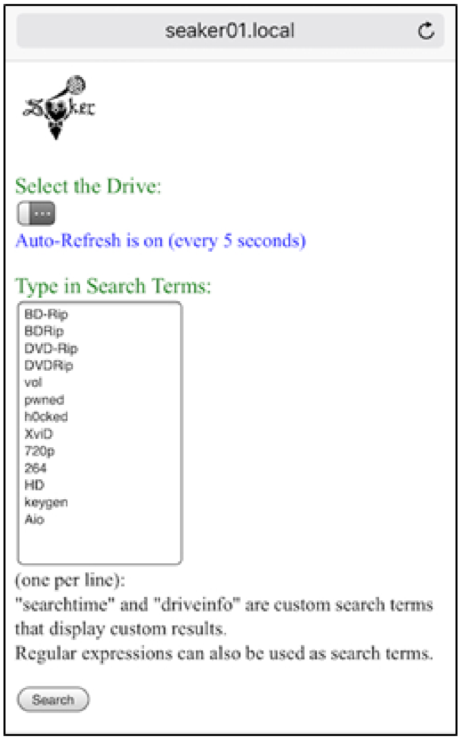
There were several options for searching for files. The simplest way is to use the built-in operating system mechanisms, for instance *ls* or *find*. Another way is to code a simple C program that does the same thing. There is another built-in operating system mechanism that is much faster when used on standard unix-based environments called *locate*, however, that utilizes an index that is built up over time and does not work well with newly attached devices.

Since one of the goals of the SEAKER project was speed of collection, a test was performed that measured the built-in mechanisms vs the simple C program. The simple C program was by far the fastest, beating the other two methods by an average of 26% (see Table 2 for data). My theory on why is that the other programs have many built-in options and functionalities that are not utilized. However, those extra functionalities have unnecessary code switches and therefore extra execution paths that are not necessary for this file and directory collection application.

3.5 Web Code

HTML and PHP were chosen as a simple interface for quick development. It also allows for easy access via many different devices through a standard web browser.

The main page (Figure 4) was designed to be very easy to use and as automatic as possible. The list of searchable devices is populated at the load-time of the page. As well, the page refreshes itself until searchable media is found. The default keyword search list is also shown on the page. Each keyword is searched independently and are entered one per line on the input keyword text form. The keyword search list is read in using PHP from a pre-determined file that resides on the Micro SD card.



Select the Drive:

Auto-Refresh is on (every 5 seconds)

Type in Search Terms:

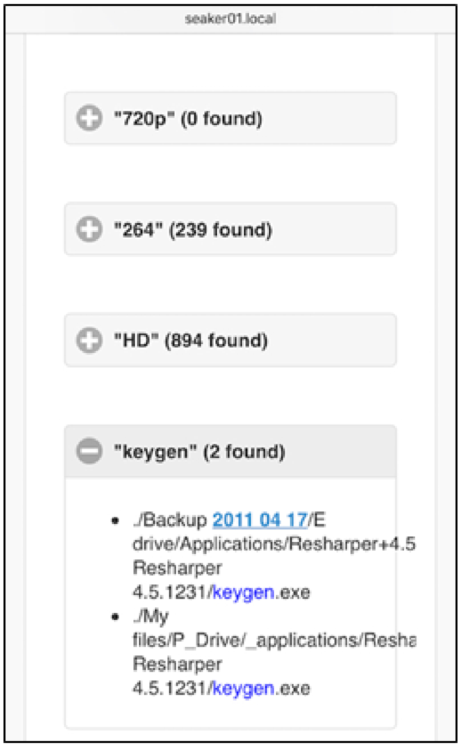
```
BD-Rip
BDRip
DVD-Rip
DVDRip
vol
pwned
h0cked
XviD
720p
264
HD
keygen
Aio
```

(one per line):
"searchtime" and "driveinfo" are custom search terms that display custom results.
Regular expressions can also be used as search terms.

Figure 4: Main Page

When triggered using a *Search* button, the media list and the keyword list to search are passed to the web server via a webpage POST. The results page is then dynamically created using PHP to read each media's file and directory list and write the matching results to the returned webpage. The Raspbian operating system built-in command *grep* is utilized to find the results on the server side.

The resulting page (Figure 5) is then displayed to the user using a simple HTML expand/collapse tool. Each media searched and each search criteria are accessible via this tree-like tool.



⊕ "720p" (0 found)

⊕ "264" (239 found)

⊕ "HD" (894 found)

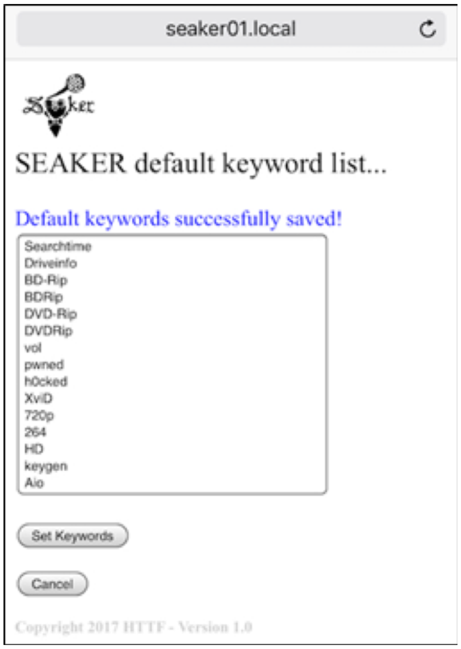
⊖ "keygen" (2 found)

- ./Backup 2011 04 17/E drive/Applications/Resharper+4.5 Resharper 4.5.1231/keygen.exe
- ./My files/P_Drive/_applications/Resha Resharper 4.5.1231/keygen.exe

Figure 5: Results Page

In addition, some special keywords can also be used to obtain information about the media. The search term *driveinfo* can be used to list the exact details of the media, including serial number, size, and other information. The search term *searchtime* can also be used to show the time needed to find the full drive information as well as the time for the particular search.

An administration page was also created for editing the default search keywords. This page was intended to be password protected, but was instead conceiled by not providing a link to it. The full path to it (http://seaker01.local/keywords.php) is required for access. Once changed, the default search keywords remain permanently altered for the SEAKER device. See Figure 6:



SEAKER default keyword list...

Default keywords successfully saved!

Searchtime
Driveinfo
BD-Rip
BDRip
DVD-Rip
DVDRip
vol
pwned
h0cked
XviD
720p
264
HD
keygen
Aio

Copyright 2017 HTTF - Version 1.0

Figure 6: Default Keywords page

3.6 Process Flow

The SEAKER device usage involves two main flows. The first is the process of *collection*, which involves plugging it in to power and then attaching digital media devices to it to be scanned. The second process is *searching*, as discussed in a previous section. See Figure 7 for details.

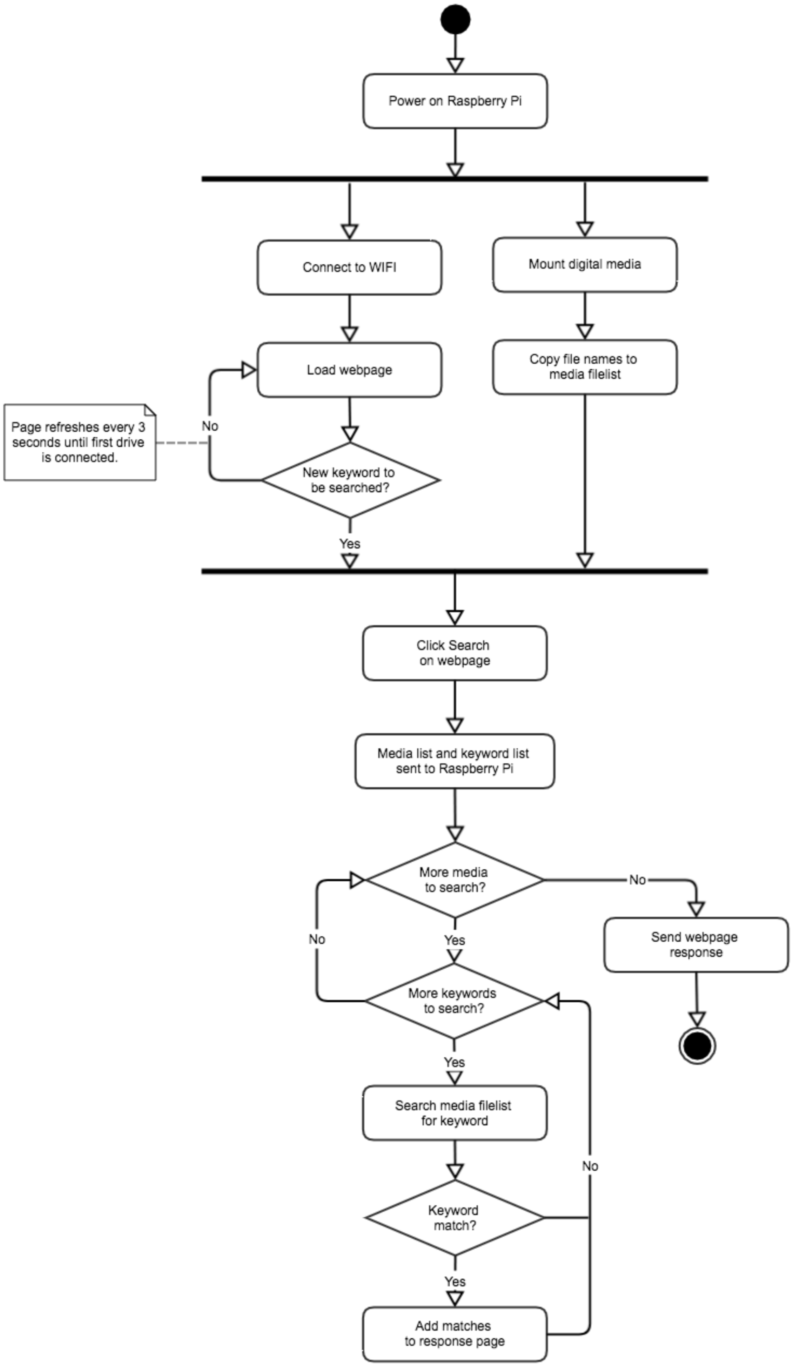

Power on Raspberry Pi
Connect to WIFI
Mount digital media
Load webpage
Copy file names to media filelist
Page refreshes every 3 seconds until first drive is connected.
No
New keyword to be searched?
Yes
Click Search on webpage
Media list and keyword list sent to Raspberry Pi
More media to search?
No
Send webpage response
No
Yes
More keywords to search?
Yes
Search media filelist for keyword
No
Keyword match?
Yes
Add matches to response page

Figure 7: General Process Flow

The process flow of SEAKER is a very simple design. After turning on the Raspberry Pi, the two processes happen simultaneously. Immediately when the drive(s) are connected to the Raspberry Pi, the file names and their paths are collected and copied to a text file. Meanwhile, the user must connect to the Raspberry Pi via WIFI on a separate wireless-enabled device. The user must then open the SEAKER web page. As shown in General Process Flow of Figure 7, the web page will refresh every 3 seconds, looking for new drives to be connected to the Pi. The first drive will be automatically added to the list of drives available on the webpage to be searched. All additional drives will appear in the list when the user refreshes the page manually.

All user selected drives will then go through the search process as shown in the lower section of the General Process Flow of Figure 7. Each drive will be processed one at a time. For example, the list of files for the drive will be scanned for any matches to the first keyword in the list. The search is done using the regular expression tool embedded in the Raspbian Linux operating system called *grep*. All files that are found to match that keyword will be added to the output HTML. Once the entire file list has been searched for that keyword, the process will begin again with the next keyword in the list. This process will continue until there are no more keywords to be searched. If multiple drives have been selected to be searched, the same process will repeat itself for each drive. Finally, the PHP engine will finish processing and the HTML response page is finalized and sent back to the user's mobile device.

3.7 Tools Used for Development

3.7.1 Hardware

(1) *Raspberry Pi and power cord* - A powerful educational and research tool that consists of an affordable, very small form factor, medium-powered computer. It follows the standard "von Neumann architecture" by incorporating three main components: input/output devices, memory, and a central processing unit. Models 3 and 3 B+ of the Raspberry Pi were used for creation and testing of the SEAKER project. More information can be found at http://raspberrypi.org.

(2) *Micro SD Card* - A solid state storage device that when combined with a Raspberry Pi becomes a local hard drive for operating system, programs, and storage while the SEAKER device is powered on. The range of Micro SD cards available are by storage size and speed of reading and writing (denoted by "class" designation). For the SEAKER device, the faster the write speed, the faster the *collection* phase happens because of having to store the results. A "class 10" or better cards have produced results more quickly than other cards with slower write speeds.

(3) *Powered USB to SATA adaptor* - These devices are faily common, and allow a SATA style hard drive to be connected to a computer through the standard USB port. This enables SATA style hard drives to be connected to the SEAKER device. It is important to note that the USB ports on the Raspberry Pi do not have enough power to support powering larger hard drives.

(4) *Router or switch* - A router or switch allows multiple computers to connect to each other using specialized software through network interface cards. During the setup phase of the SEAKER device, one of these is necessary to be able to

connect to Raspberry Pi with another computer. Remotely logging into the Raspberry Pi and running the setup script is a required step (see setup steps in the appendix).

(5) *Ethernet cable* - This is a hard-wire cable designed to connect a computer to a network. Since the setup process alters the wireless network interface card, wireless setup is not an option for the SEAKER device.

3.7.2 Programming Languages and Scripting

(1) *bash* - A unix-based shell, command language, and shell scripting language used to gather and run a set of operating system commands together. Bash is used in several places in the SEAKER project, including for the setup script, as an auto-mount script for automatic *collection* and in the PHP code to obtain and push search details to the output HTML.

(2) *c* - The c programming language is a mid to high level language for writing computer programs. This was used in two locations to provide programs for media content *collection* and for a late added feature to control a light-bar of LEDs that plugs into the GPIO port of the Raspberry Pi.

(3) *HTML* - A tag (or element) based system of organizing information to be shown in a web browser.

(4) *CSS* - An HTML styling language that simplifies the ability to modify how HTML is displayed in modern web browsers.

(5) *PHP* - A server-side programming langauge, typically associated with dynamic web page creation. For the SEAKER project, this was a vital component of creating the search and results web pages.

(6) *JavaScript* - A client-side programming language that enables web developers to make web pages more interactive without having to access server-side resources.

(7) *JQuery* - A standard JavaScript library of useful functions to simplify and expand the ability to make a web page more interactive.

(8) *Regular Expressions via grep* - Regular expressions are a complex way to specify a search or replace pattern. *grep* is a unix-based operating system command-line utility that is used to search for regular expression patterns in plain-text files. This is a very critical part of the SEAKER device, since the searching mechanism plays a vital role in the project.

3.7.3 Raspbian Operating System

(1) *Raspbian Linux (a Debian distro)* - One of the standard operating systems that support the Raspberry Pi hardware. It is derived from the Debian operating system. There are two versions available for every release: Full and Lite. The full version provides lots of useful default packages, a built-in user interface and a very simple to use guided setup. The lite version provides only a minimum of packages, no built-in user interface (with the exception of a command line), and no initial guided setup. In order to keep the SEAKER project as uncomplicated as possible, the lite version of the Raspbian operating system was chosen.

(2) *grep* - A command-line utility included with almost every unix-based operating system. It is widely known as a very powerful tool for finding regular expression patterns in plain-text files. In the Raspbian operating system, the initial configuration contains some useful features that colorize the results of grep.

Unfortunately, this also slows down the grep mechanism due to the extra code burden. The SEAKER setup script removes the color features of grep to enable the fastest searching experience. It also changes the default searching mechanism to ASCII search, enabling an even faster grep result.

(3) *apt-get for os packages* - Advanced Packaging Tool packages are automatically searched, downloaded, and installed using the apt-get command-line utility. It is a tool based on the Debian software packaging system. The SEAKER project uses this tool inside the setup script to add necessary packages to the Raspbian lite operating system.

(4) *Apache HTTP server* - An industry standard web server application that is open source and cross platform.

(5) *PHP add-on for Apache* - Hypertext Preprocessor utility that enables a programming language for server-side coding.

(6) *Rules file for auto-mounting* - The *udev* utility is a userspace device manager for Linux-based operating systems. One mechanism of device control is to write a simple script file called a "rules" file (similar to a shell script) that enables the device manager to automatically control the specified device(s). Because the SEAKER project intended to auto-mount digital media and launch a *collection* script to gather the information from it, a rules file was created during the setup script that enables the device manager to automatically perform these tasks.

3.7.4 Collaboration Tools

(1) *Gliffy* - A free Google Chrome application that enables diagramming charts, graphs, flowcharts, and other drawing templates. This tool has a simple drag

and drop interface that allows for online collaboration and sharing during diagram creation. The SEAKER project participants utilized this for creating flowcharts describing the user and process flows.

(2) *Slack* - An online instant messaging and group chatting platform freely available for simple use. The SEAKER project participants utilized this for collaborating, setting up meetings, online video discussions, and some initial document sharing.

(3) *AWS S3* - Amazon Web Services' Simple Storage Service (S3) is for online storage of files. The SEAKER project utilized this for beta versions of the setup script and eventually storage of the final working version along with the documentation to accompany it.

(4) *GitHub* - An online code sharing, repository, and version control system. It is widely known to host many open source projects and repositories. The SEAKER project used it during development to share the code base and create an informational readme file that is displayed in *GitHub* on the main repository page.

(5) *Dropbox Paper* - An online document repository and collaboration tool that enables multiple people to edit the same document simultaneously. The SEAKER project team utilized this tool for writing up the setup guide, user guide, technical specification, and presentations.

3.7.5 Setup Tools

(1) *Hardwire Connection to the Internet* - The SEAKER setup script requires a connection to the internet in order to download the necessary operating system packages. The ideal SEAKER setup environment is to connect the Raspberry

Pi to a router or switch that is also connected to the internet without a proxy server. This allows another computer to securely connect to it as well as having a connection to the Internet.

(2) *Other Hardware* - Please see the Hardware section earlier in this chapter.

(3) *Apple Computer System: Etcher, Paragon NTFS, terminal, ssh* - The secondary computer required to remotely access the Raspberry Pi in order to run the setup script can be an Apple computer or a Windows computer. In order to load the initial Raspbian lite operating system onto the Micro SD card, a program like *Etcher* can be used. It is a tool for copying a disk image from a file to a Micro SD card. Another necessary tool is an NTFS reader/writer driver like *Paragon NTFS* in order to access the newly copied image of Raspbian lite on the Micro SD Card. This is necessary to copy the setup script and enable SSH access to the Raspberry Pi. Finally, a command line program like *terminal* can be used to remotely and securely connect to the Raspberry Pi after the Micro SD card is installed and the system is booted.

(4) *Windows Computer System: Win32DiskImager, Windows Explorer, Putty* - The secondary computer required to remotely access the Raspberry Pi in order to run the setup script can be a Windows computer or an Apple computer. In order to load the initial Raspbian lite operating system onto the Micro SD card, a program like *Win32DiskImager* can be used. It is a tool for copying a disk image from a file to a Micro SD card. Another required action is to access the newly copied image of Raspbian lite on the Micro SD Card. This is necessary to copy the setup script and enable SSH access to the Raspberry Pi. Finally, a SSH program like *Putty* can be used to remotely and securely connect to the Raspberry Pi after

the Micro SD card is installed and the system is booted.

4 Experimental Results

4.1 Prototype Demonstration

As the final project in the summer 2017 Cyber Security class, the SEAKER device project was presented to the SCHTTF representatives and other faculty and administration from CSUCI.

The SEAKER device presentation consisted of an introduction from Dr. Soltys, a slide presentation from every student team, a recommendation for hardware to use for real-world SEAKER implementations, and a live demonstration of SEAKER's functionalities with prepared, never-before-attempted digital media provided by the SCHTTF. The introduction and slide presentations were explanatory in nature and provided a good basis of understanding for the live demonstration.

As ambitious as the project was, the live demonstration was intended to be fully functional and prove that a prototype could be created using the described parameters and features requested. The audience was also encouraged to participate as "detectives" using their personal phones to experience the SEAKER functionality first hand. As well, an Apple iPad was projected onto the presentation screen at the front of the room.

After the prototype of the SEAKER device was up and running, an initial test with a known tested digital media device was performed. After attaching it to the SEAKER device, the auto-mount script was run and the information was properly collected and searchable. The auto-refresh functionality on the webpage worked as expected and showed the attached media as a selectable drive. A quick search also was successful and the product was demonstrated without flaw.

Finally, the SCHTTF prepared digital media was presented in the form of three different devices: two USB storage drives and a SATA hard drive. All three devices were able to be auto-mounted and searched. The search criteria was also provided live after the drive information was collected. The final phase was to perform the search and see if the proper fileset was exposed in the results. The conclusion of the live demonstration was that the fileset was properly found and the SEAKER device was properly working and ready to experiment more in a real-world situation.

4.2 Results

TODO:

graph of latency

estimates of time vs data Gb

examples of running on Frank disks

5 Conclusions and Future Work

SEAKER is currently a prototype; there are many improvements to be made, and we will discuss some of them in this section. These improvements may be implemented by future students, or by digital forensics professionals. We encourage anyone who implements them to share their work; the main bash script for the SEAKER device is available on github at https://github.com/michaelsoltys/seaker.

First, SEAKER supports a select few filesystems. Namely, NTFS and FAT (exfat, FAT32, FAT16...). There are likely bugs to be worked out in the supported filesystems, and there is certainly work to be done in expanding the list of supported systems.

If an unsupported filesystem is in use, it may simply fail to mount, and not show up on the SEAKER site at all. Similarly, drives from which files are currently being collected do not appear; the site displays them only when files have been collected. Here there is opportunity for improvement; as opposed to displaying drives for which collection is complete, SEAKER could display all drives, and a status next to each. This status only requires three states: failed search (for unsupported systems), collection in progress, and collection complete.

It can be difficult to match a hard drive to its corresponding search results. Partitions are uniquely identified by a UUID, and some properties (capacity, for example) are displayed with the search results, but these properties do not provide a perfect way to determine which physical device corresponds to which mounted partition or device. Storage devices generally have a serial number of sorts, but this serial number is not visible to SEAKER. This is a problem which requires some creativity to solve well. SEAKER could take a picture when a drive is plugged in, and associate that picture with the search results, for instance, but this solution requires that investigators position each storage device in front of a camera; this approach requires a camera, and moreover it is tedious and error-prone.

When a search finds a hit (i.e., a matched expression or file extension), investigators may want to view the corresponding file. Currently, this would require them to manually find and open the file. Speed and ease of use are priorities, so it would be best if investigators could select a file in the search results and have SEAKER fetch a copy of it for them. This function inevitably requires that the storage device being queried is still connectedassuming that this condition is met, copying and viewing a file should not be too complex.

Similarly, it would be useful if investigators could view thumbnails of images and videos in the search results. One example of the motivation here is child pornography cases; incriminating images may have innocuous names, but thumbnails would indicated the true content.

This leads to another issue: as incriminating files may be named innocuously, investigators will often want to search simply for all images, videos, etc. SEAKER could minimize the work necessary by allowing for preset groups of search terms, which can be created and edited by administrators. For example, an admin could create an images group which causes SEAKER to include jpg, pdf, png...

We are very interested in a Data Carving option. Data carving is the identification and extraction of files from unallocated clusters using file signatures. A file signature, also commonly referred to as a magic number, is a constant numerical or text value used to identify a file format. The object of carving is to identify and extract (carve) the file based on this signature information alone. We are interested in hidden files (which are sometimes easy to locate, as for example in UNIX with *ls -a* command) and deleted files, which is more tricky as the files can partially overwritten. A partially overwritten file may still constitute valuable evidence: for example, a portion of an image can be taken as solid evidence that the entire image was on the disk at some point. How can one establish whether a portion of an image comes from a particular image? It seems that the only way to accomplish that is by visual inspection, and having an investigator recognize the original image. In order to automate this process one could attempt one of two things: build a massive database of frequently circulating (say, CP) images, and hashing different formats of these images (.pdf, .jpg, .giff, .tiff, etc.), as well as different resolutions, and chunks of standard sizes (say, 64Kb). This still seems like a shot in the dark. The second approach is to define something akin to fuzzy hashes, the type of hashes that are used to recognize variants of the same malware. This new type of fuzzy hashing would be invariant under different formats, or standard resolutions, and chunks of an image could be identified by close proximity to the original hash. Hits would be still confirmed visually to avoid false positives; a bigger issue would be false negatives.

Finally, documentation is important in any investigation. When triage reveals media which motivates investigators to confiscate the corresponding storage device, they should document this motivation. As such, it would aid investigators if SEAKER could generate a search report for a selected drive from the search results screen. This report could be downloaded to the investigators device or saved on the SEAKER unit for later access by an administrator. It should contain the search results along with some circumstantial information, such as the date, the name(s) of investigator(s) requesting the report, and their reason for confiscating the device.

The potential uses for the SEAKER device are great with the existing set of functionality. However, the future potential functionalities are even greater.

The general implementation and code have some limitations. For instance, in addition to the regular expressions, there could also be fuzzy matching, size grouping, internet browser data, registry, swap file, and email searching, and many others.

Another potential area for improvements to existing code could be a location for entering passwords obtained from suspects. These could be used to unlock entire drives, zip files, PDFs, user folders, files, email, website usage, etc. This could also be extended to find online account passwords, especially in the case of browsers that allow saving site-specific usernames and passwords.

The SCHTTF has asked for some new functionality as well. They would like the ability to search multiple partitions, local WIFI networks and access levels, thumbnails of images and videos, support newer operating systems (APFS and HFS Plus), and extract IP addresses from known suspect configuration files and other locations. These are just a few of the requests, but seemed to be at the top of their list.

As well, there are many different libraries that could be included and built into a searching algorithm. Be aware that these will slow down the gathering process and could be more useful if searched in stages. Here are a few:

- LibForensics http://code.google.com/p/libforensics/
- Volatile Memory search (Volatility: http://code.google.com/p/volatility/) and (WindowsSCOPE: http://www.windowsscope.com/)
- Oxygen for mobile http://www.oxygen-forensic.com/en/features

In researching this topic a lot of other potential improvement features came to mind. This is by no means a comprehensive list, but it is a start:

- More supported media types
- Add AJAX (or similar technology) to give live feedback for search and collection

- Create RESTful API for using Raspberry Pi
- Clean up HTML code, use CSS
- Write SEAKER iPhone/iPad/Android apps to connect to Raspberry Pi and perform searches, edit keywords, etc.
- Use heap memory for path
- Better way to skip . and .. (possibly always skip first two entries?)
- Reduce size of file/directory listing file. This may involve changing how to grep or implementing custom grep
- Possibly store files in database instead of a file
- Implement the ability to connect to Raspberry Pi using Bluetooth instead of WIFI
- Customize web pages for device type
- Use a better wireless network adapter for better range. These adapters are inexpensive.
- Implement Linux setup with puppet/cfengine/salt/etc
- Better error messages about why drive could not be read
- Add a Blinkt! light panel to show visual status of the Raspberry Pi
- Check the health (SMART status) of the hard drive before scanning
- Add support for RAID, mSATA, SCSI hard drives (mdadm)

- Read directly from rawdisk to find file list to speed up collect
- Make SEAKER an available Raspbian/Debian package
- Support unicode filenames
- Auto-unmount the hard drive at the end of collection
- Support multiple partitions gracefully
- Search for filename matches only
- Search for path matches only
- Offer option via checkbox for searching inside compressed files
- Offer option via checkbox for searching inside text files
- Find all deleted files (foremost, ntfsundelete)
- Search deleted partitions and unpartitioned space
- Build another web page for troubleshooting/access/administration/etc
- Search on-media virtual hard drives (vhd, vdi, vmdk) (vmware, virtual pc, parallels, hyper-v)
- Search on-media hard drive images (.iso)
- Searching the raw drive instead of using the on-media operating system
- Online hard drive investigation (i.e. Cloud Forensics)
- Network Traffic Investigation

- Video segmentation and video image hashing
- Crime-specific searchs:
  - financial crimes
  - credit card fraud
  - hacking
  - bullying
  - bloackmail
  - espianage
  - fraud
  - customizable (corporate / military)
- OS lockdown (raspbian)
- Decrypting encrypted devices (password entry location, assessment without password)
- Utilize forensics as a service
- Integrate with Microsoft's photoDNA cloud service
- Build an iPad app to simpler, more guided use
- Query Expansion - automatically searching for same query maybe other contexts
- Synonym Matching - automatically searching for similar words to query word

- Collect everything in UTC time for chronology matching
- Universal way of collecting hard drive hash for verification of evidence integrity
- Data Visualizations:
  - present all data visualizations for particular drive or all hard drives
  - graph - size vs amount of files (one hard drive, and all hard drives)
  - graph - common details (like file type, etc) maybe clickable!
  - graph/chart - files by date
  - graph/chart - files by file type
  - chart - website visits
  - digital image hashes list (stored and compared)
  - many others...
- Improve analysis speed: skip known OS files, known applications files, etc.
- Investigation Gathering rollup: (possibly stored online or in a report)
  - Database Schema for storing case specific data
  - metadata
  - Unique "gathering ID"
  - case number
  - observation report
  - crime severity

  - potential offenses
  - time gathered
  - gatherer
  - suspect list
  - location gathered
  - suggestions for other research
  - which computer system it came from
  - Set of evidence
  - Digital Evidence item
  - images of item
  - unique item ID
  - file contents
  - ranking within set of evidence
  - image thumbnails
  - collection statistics
  - etc

- Find encryption Keys
- Thumb strips of videos
- Predetermined search criteria (passwords, pw, etc)
- Output more file information: file owner, MAC times

- Sorting ability, for instance based on user or access times
- Ability to search by time, i.e. time=lastweek, time=5/5/18-5/15/18
- Internet usage timeline
- Auto-search / Auto-filter
- For drug related crimes, search... Spreadsheets, documents, databases, internet purchase strives
- For financial related crimes, search... Spreadsheets, databases, MSMoney, Quicken
- CRC of any acquired files (for later integrity comparison)

The SEAKER project was a successful collaboration between two different institutions in the public sector: law enforcement and academia. The former has many interesting problems to offer, but as they are overwhelmed with cases they typically do not have the man power to do research and development. The latter is happy to do research and development, as it enhances the educational experience of the students to be learning in the context of applications to real life problems. It is a fortuitous and symbiotic relationship, and we plan to embark on other such projects in the future.

SEAKER is also the testament to the fact that supremely useful devices, meeting the needs of practitioners, can be constructed from relatively simple components; what is required is expertise and enthusiasm, which in the best cases academia possesses in ample measure. RPs are a revolution in embedded controllers, and we are just scratching the surface of their applicability. They are inexpensive, but wield the power of the Linux OS.

For the students, the experience was invaluable. Perhaps the most important aspect was non-technical: how to work well in a large team. There were eighteen students in the group; a composition of different backgrounds, talents and strengths. We divided the task into five different but interconnected teams: Task 1 was connecting the external devices to the RP; Task 2 was searching the contents of the devices; Task 3 was responsible for sending the query and retrieving the results of the search to the handheld; Task 4 was responsible for the documentation of the project (both a user set up and guide, as well as the technical documentation of the solution); Task 5 was responsible for testing.

Digital forensics and academia would both benefit greatly from increased collaboration; students can offer relatively inexpensive development in exchange for real-world experience and the opportunity to create something which will be used. As a side effect more students would consider digital forensics as a career, resulting in some level of alleviation of the problems mentioned in the second quote in the introduction[5].

6 Appendix

6.1 SEAKER Setup

The following set of instructions will detail how to setup the SEAKER environment for the first time. There are three install options that enable SEAKER creators to prepare the device. See Table 1

1. *Router*: This option is where the Raspberry Pi and the secondary computer are connected directly to the same router, thus allowing the same local DHCP to assign the IPs of both. The secondary computer is used to prepare the micro SD card and to later remotely and securely connect to the Raspberry Pi to complete the setup.

2. *Direct Connect*: This option is where the Raspeberry Pi is connected directly to a monitor and keyboard to enable direct user input via the termnial. The secondary computer is necessary to prepare the micro SD card, but not used to remotely connect to the Raspberry Pi to complete the set up.

3. *Corporate LAN*: This option is almost identical to the *Router* option, but utilizes a corporate network instead of a local router to connect to the Raspberry Pi. This option is the most IT intensive, since the IP address assigned to the Raspberry Pi is often not easily found.

Option 1 **(Router)**	**Option 2** **(Direct Connect)**	**Option 3** **(Corporate LAN)**
Hardware Required		
•Raspberry Pi •Mac or Windows Computer •Micro SD card •Router	•Raspberry Pi •Mac or Windows Computer •Micro SD card •Monitor •Keyboard	•Raspberry Pi •Mac or Windows Computer •Micro SD card
Software Required		
•Raspian Stretch Lite •ssh client •Disk Imaging software •IP Scanning software	•Raspian Stretch Lite •ssh client •Disk Imaging software	•Raspian Stretch Lite •ssh client •Disk Imaging software

Table 1: SEAKER set up: Required Hardware and Software

The following creation process is required to setup the SEAKER device to the specifications outlined in this paper. Figure 8 outlines the general process, while the specific steps are listed below.

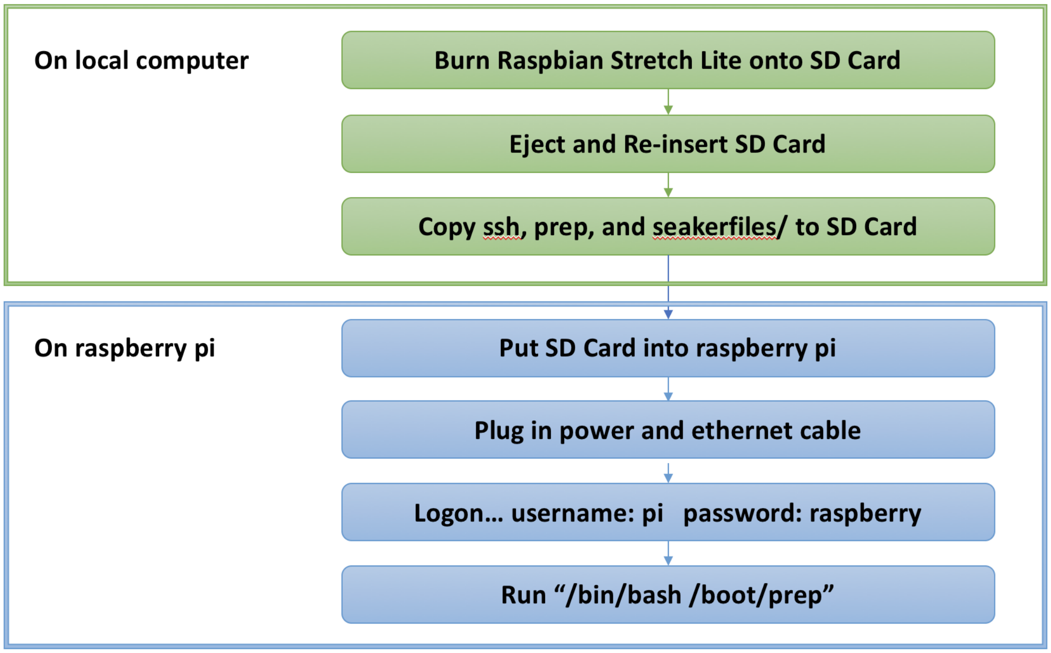

On local computer
Burn Raspbian Stretch Lite onto SD Card
Eject and Re-insert SD Card
Copy ssh, prep, and seakerfiles/ to SD Card
On raspberry pi
Put SD Card into raspberry pi
Plug in power and ethernet cable
Logon… username: pi password: raspberry
Run "/bin/bash /boot/prep"

Figure 8: SEAKER Creation Process

1. Download the latest Raspbian Stretch Lite operating system (https://www.raspberrypi.org/downloads/raspbian/). Note the location where file is saved.

2. Download the most recent copy of *prep.sh*, *ssh*, and the folder called *seakerfiles*. These files contain SEAKER setup and running code.

    - prep.sh file location: https://s3-us-west-2.amazonaws.com/seaker/prep.sh
    - ssh file location: https://s3-us-west-2.amazonaws.com/seaker/ssh

- seakerfiles location: https://s3-us-west-2.amazonaws.com/seaker/seakerfiles

3. Open prep.sh and edit the default configuration information (shown below). At minimum it is recommended to change the Raspberry Pi and WiFi passwords.

```
# CONFIGURATION SETTINGS
# Raspberry Pi Password
PI_PASSWORD=”raspberry”
# WiFi Network Name
WIFI_NAME=”SEAKER01”
# WiFi Network Password:
WIFI_PASSWORD=”raspberry”
# IP address which is used to access SEAKER web page
WIFI_ROUTER_IP=”192.168.101.1”
# DHCP Range (How many connections can be made simultaneously)
WIFI_ROUTER_DHCP_RANGE=”192.168.101.50 192.168.101.100
```

NOTE: It is recommended to change the WIFI Name and IP Address when setting up multiple SEAKER environments over time to ensure each environment has unique identifying information.

For example: If setting up three SEAKER environments, configuration could be:

(a) Name: SEAKER01, IP Address: 192.168.101.1

(b) Name: SEAKER02, IP Address: 192.168.102.1

(c) Name: SEAKER03, IP Address: 192.168.103.1

4. Insert micro SD card into computer (not the Raspberry Pi). An adapter will likely be required.

5. Open disk imaging software (Etcher for Mac, or SDFormatter and Win32 Disk Imager for Windows). Map to the Raspbian Stretch Lite file location, choose the micro SD card as the destination, and select to burn the image. (Refer to the chosen imaging software documentation for specific instructions on using this tool.) Do not remove the micro SD card from the computer.

6. Map to the micro SD card (Finder for Mac, File Explorer for Windows). Copy ssh, prep.sh, and the seakerfiles folder onto the micro SD card's *boot* partition.

7. Remove the micro SD card from the computer.

8. Insert the card into the Raspberry Pi.

9. Power on the Raspberry Pi by plugging it in with the power cord.

10. Identify the local IP Address of the Raspberry Pi:

    If installing with Option 1 (router):

    (a) Plug the Raspberry Pi into the same router being used by the Windows or Mac computer.

    (b) Use the IP Scanning tool on the computer to find the local IP Address of the Raspberry Pi. The Manufacturer should be Raspberry Pi Foundation.

    If installing with Option 2 (Direct Connect):

    (a) Connect the monitor and keyboard to the Raspberry Pi.

    (b) Login using the default username (pi) and password (raspberry).

(c) Enter the following command to retrieve the local IP Address:

```
ifconfig eth0
```

If installing with Option 3 (Corporate Network):

(a) The MAC Address of the Raspberry Pi is required. This can be located on the original Raspberry Pi box.

(b) For Windows systems, open a command prompt and enter the command below. Replace the "c8:26:3b:d2:63:d5" sequence with the MAC Address of the Raspberry Pi being configured. Use the following command:

```
arp -a | findstr "c8:26:3b:d2:63:d5"
```

(c) For Unix or Linux systems such as Apple or Ubuntu, open a terminal window and enter the command below. Replace the "c8:26:3b:d2:63:d5" sequence with the MAC Address of the Raspberry Pi being configured. Use the following command:

```
arp -a | grep "c8:26:3b:d2:63:d5"
```

11. If installing with Option 1 or 3:

- SSH into the Raspberry Pi from the laptop or desktop computer.

  - If using a client such as Putty, enter the local IP address of the Raspberry Pi, choose SSH and connect. Click `OK` or `Yes` on the security warning.
  - If using a command line utility such as Bash enter the following at the prompt:

```
ssh pi@<ip_address> -l pi
```

  - Login using the default username (`pi`) and password (`raspberry`).

12. Run the preparation script by typing the following on the command line:

```
/bin/bash /boot/prep.sh
```

13. Wait for the Raspberry Pi to finish running the script and rebooting. The Raspberry Pi should now be configured as a SEAKER and be up and running.

6.2 SEAKER Usage

After collecting all the media devices at the scene, the investigator will triage them with SEAKER. Each step of the process is broken down and discussed in detail below.

(1) Connect the RP to the power and wait for about a minute to let it finish booting up. Connect to the RP's Wireless Access Point (WiFi network). Depending on the setup, the WiFi's SSID will be "SEAKER01" or "SEAKER02" etc. See Figure 9. Note that SEAKER's Wireless Access Point is password protected and matches the one specified in the prep.sh script file.

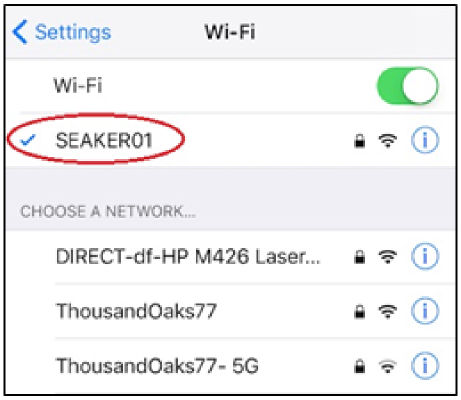
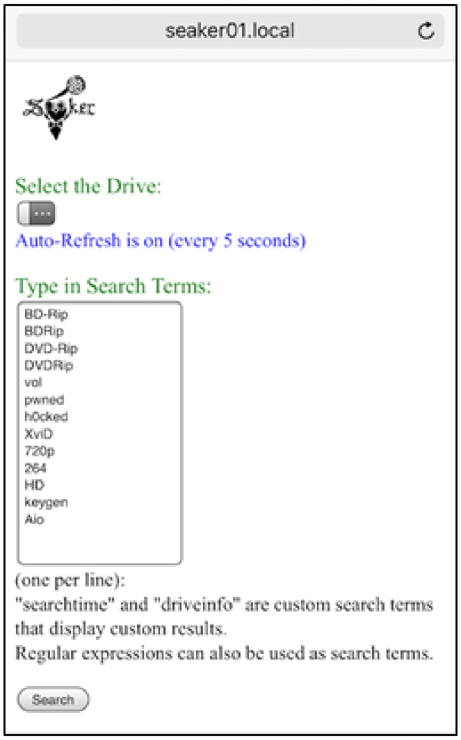


Figure 9: iPhone WIFI connection to the SEAKER.

(2) At the same time the investigator may connect all the media devices to the RP. This may be done concurrently with the previous step. Note that in order to examine a digital media it will need to be removed from the computer, and connected to the RP; this may be done through a write-blocker interface but it is

not necessary.

(3) Once connected to the SEAKER's Wireless Access Point, the investigator will open any web browser on their connected device and direct it to go to `http://seaker01.local`. Access is allowed through a web browser, as this is the most universal way to connect on any device (iPhone, iPad, Android, laptop, etc.). These devices and many more can connect to a Wireless Access Point and open a browser. Once the browser establishes the connection, the user will see Figure 10. Note that the keywords (or regular expression patterns) present in the "Type in Search Terms:" can be pre-loaded before arriving at the scene, or changed/updated at the scene.



Select the Drive:

Auto-Refresh is on (every 5 seconds)

Type in Search Terms:

```
BD-Rip
BDRip
DVD-Rip
DVDRip
vol
pwned
h0cked
XviD
720p
264
HD
keygen
Aio
```

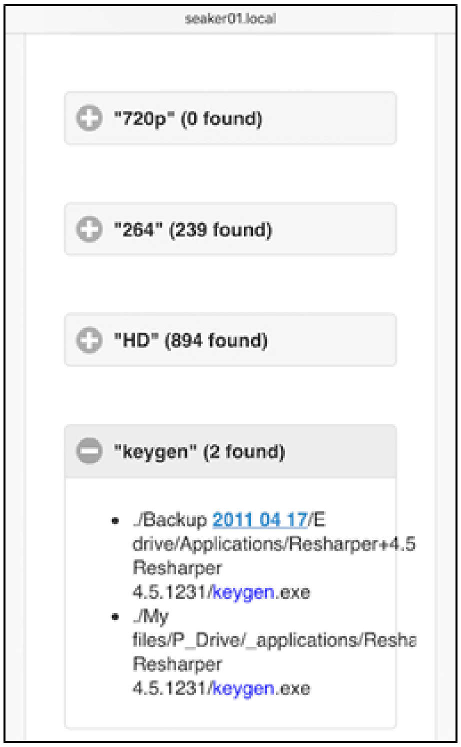
(one per line):
"searchtime" and "driveinfo" are custom search terms that display custom results.
Regular expressions can also be used as search terms.

Figure 10: Using a browser to connect to `http://seaker01.local`

The regular expression can be given using the syntax of the `grep` utility. For example, if we want to find occurrences of either ‘two’ or ‘too’, we use `t[wo]o`; if we want to find every word that start with capital letters, we use `^[A-Z]`; if we want to find words where number 9 is the last character of the line, we use `9$`. There are a vast number of possibilities; we can also replace `grep` with `egrep` that has an even richer syntax.

(4) Once any storage media devices that are found at a search warrant scene are connected to the RP, the investigator will typically wait for a few minutes (with some times up to 10 minutes for 1Tb disks with millions of files) for the file

list to be built. Searches can then be carried out very quickly; essentially, `grep` browses the file list, line by line, outputting those lines that conform to at least one pattern specified in the “Type in Search Terms:” window. Once this finishes, the investigator will have the results presented as in Figure 11.



seaker01.local

"720p" (0 found)

"264" (239 found)

"HD" (894 found)

"keygen" (2 found)

- ./Backup 2011 04 17/E drive/Applications/Resharper+4.5 Resharper 4.5.1231/keygen.exe
- ./My files/P_Drive/_applications/Resha Resharper 4.5.1231/keygen.exe

Figure 11: The results of the search of a particular device.

The filenames themselves can be incriminating evidence, such as in Child Pornography (CP) cases, where the material has a commonly used naming convention, e.g., “lolita” which can be found with the grep pattern `.*lolita.*` (‘`.*`’ means the following: ‘`.`’ (period) matches any single character of any value, except a newline, and ‘`*`’ (asterisk) matches zero or more of the preceding character or expression) or simply `lolita`. This can be used by the investigators to question the suspects. The questioning usually takes place at the same time as the forensic examiners triage the evidence, and one of the requirements of SEAKER was to be fast so that investigators can start getting intelligence quickly from the initial processing of the scene.

(5) The user process flow is documented in the following figure 12.

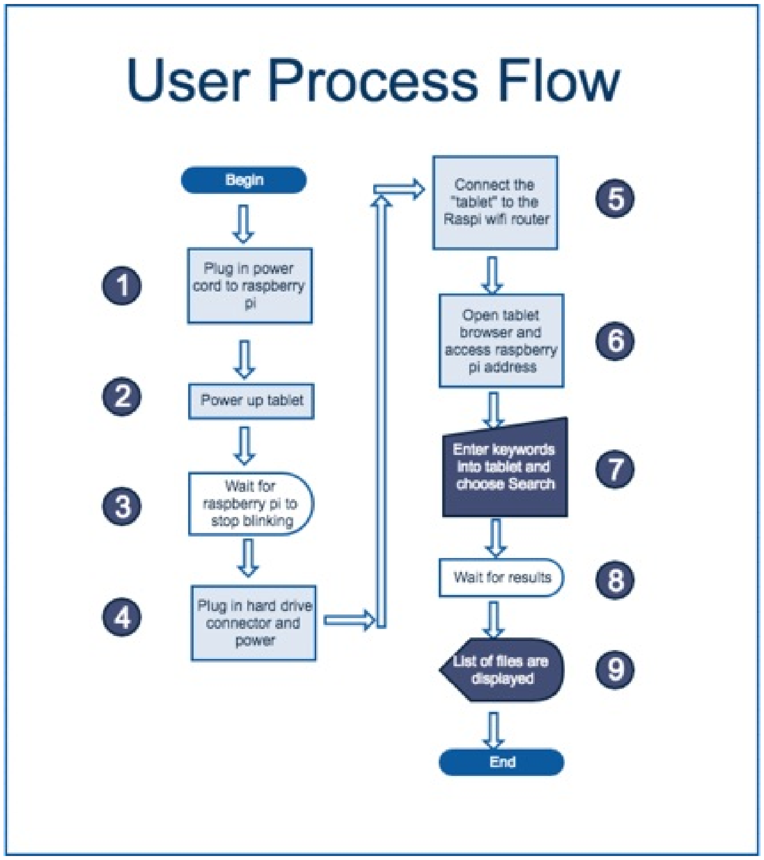

User Process Flow
Begin
1
Plug in power cord to raspberry pi
2
Power up tablet
3
Wait for raspberry pi to stop blinking
4
Plug in hard drive connector and power
5
Connect the "tablet" to the Raspi wifi router
6
Open tablet browser and access raspberry pi address
7
Enter keywords into tablet and choose Search
8
Wait for results
9
List of files are displayed
End

Figure 12: The functionality of SEAKER from the user perspective.

6.3 Code

6.3.1 Directory and Filename Collection Code (in C)

```c
#include <unistd.h>
#include <sys/types.h>
#include <dirent.h>
#include <stdio.h>
#include <string.h>

void listdir(const char *name)
{
  DIR *dir;
  struct dirent *entry;
  if (!(dir = opendir(name)))
  {
    return;
  }
  while ((entry = readdir(dir)) != NULL)
  {
    if (entry->d_type == DT_DIR)
    {
      char path[1024];
      if (strncmp(entry->d_name, ".", 1) == 0 || strncmp(entry->d_name, "..", 2) == 0)
      {
        continue;
      }
      snprintf(path, sizeof(path), "%s/%s", name, entry->d_name);
      listdir(path);
    }
    else
    {
      printf("%s/%s\n", name, entry->d_name);
    }
  }
  closedir(dir);
}

int main(void)
{
  listdir(".");
  return 0;
}
```

6.4 Results of Testing

6.4.1 Collection Timing

For testing purposes, ls was optimized to utilize as few time-consuming options as possible, including the *-f* option that prevents sorting and *-A* for skipping current and parent directories (. and ..) in the results. Here are the actual functions used:

```
sudo sh -c 'cd /mnt/usb && time ls -ARf1 > ~/ls_time.txt'

sudo sh -c 'cd /mnt/usb && time find / -print > ~/find_time.txt'

sudo sh -c 'cd /mnt/usb && time ~/collect > ~/collect_time.txt'
```

Testing results for the custom collection code vs. operating system file and directory listing applications:

	SSD	SSD	WD 2.5’	iOmega 3.5’	Samsung 3.5’	
			SATA HDD	IDE HDD	SATA HDD	
Size GB	500	500	500	1000	1000	
Consumed GB	94.22	239.83	456	474.2	316.6	
% Consumed	18.84%	47.97%	91.20%	47.42%	31.66%	
# files	1,244,699	1,561,132	14,487	216,356	21,556	
# directories	254,473	317,876	145	10,603	2,222	
ls t1	62.142	112.627	0.942	8.398	1.400	
ls t2	62.025	116.978	0.899	8.252	1.375	
ls t3	68.376	115.833	0.903	8.229	1.324	
ls ave time (secs)	**64.181**	**115.146**	**0.915**	**8.293**	**1.366**	
find t1	43.914	90.693	1.004	9.090	1.733	
find t2	44.958	97.619	1.014	8.315	1.667	
find t2	43.214	96.499	1.015	8.347	1.672	
find ave time (secs)	**44.029**	**94.937**	**1.011**	**8.584**	**1.691**	
col t1	20.893	68.270	0.834	7.541	1.281	
col t2	18.103	82.091	0.832	7.548	1.260	
col t3	19.381	84.823	0.804	7.531	1.248	
collect ave time (secs)	**19.459**	**78.395**	**0.823**	**7.540**	**1.263**	**Average:**
% faster than ls	70%	32%	10%	9%	8%	**26%**
% faster than find	56%	17%	19%	12%	25%	**26%**

Table 2: Collection Algorithm Timing Data

TODO: add more result data

References

[1] Akinola Ajijola, Pavol Zavarsky, and Ron Ruhl. A review and comparative evaluation of forensics guidelines of nist sp 800-101 rev. 1: 2014 and iso/iec 27037: 2012. In *Internet Security (WorldCIS), 2014 World Congress on*, pages 66–73. IEEE, 2014.

[2] Susan Ballou. *Electronic crime scene investigation: A guide for first responders.* Diane Publishing, 2010.

[3] Venansius Baryamureeba and Florence Tushabe. The enhanced digital investigation process model. In *Proceedings of the Fourth Digital Forensic Research Workshop*, pages 1–9, 2004.

[4] Brian Carrier, Eugene H Spafford, et al. Getting physical with the digital investigation process. *International Journal of digital evidence*, 2(2):1–20, 2003.

[5] Ben Hitchcock, Nhien-An Le-Khac, and Mark Scanlon. Tiered forensic methodology model for digital field triage by non-digital evidence specialists. *Digital Investigation*, 16:S75–S85, 2016.

[6] David Lillis, Brett Becker, Tadhg O'Sullivan, and Mark Scanlon. Current challenges and future research areas for digital forensic investigation. *arXiv preprint arXiv:1604.03850*, 2016.

[7] Michael G Noblett, Mark M Pollitt, and Lawrence A Presley. Recovering and examining computer forensic evidence. *Forensic Science Communications*, 2(4), 2000.

[8] Sriram Raghavan. Digital forensic research: current state of the art. *CSI Transactions on ICT*, 1(1):91–114, 2013.

[9] Marcus K Rogers, James Goldman, Rick Mislan, Timothy Wedge, and Steve Debrota. Computer forensics field triage process model. In *Proceedings of the conference on Digital Forensics, Security and Law*, page 27. Association of Digital Forensics, Security and Law, 2006.

[10] Adrian Shaw and Alan Browne. A practical and robust approach to coping with large volumes of data submitted for digital forensic examination. *Digital Investigation*, 10(2):116–128, 2013.

[11] J Williams. ACPO good practice guide for digital evidence. *Metropolitan Police Service, Association of chief police officers, GB*, 2012.

[12] Charles L Yeschke. *The art of investigative interviewing: A human approach to testimonial evidence, Second Edition.* Butterworth-Heinemann Boston, 2003.

7 STOP HERE - Abstract

8 Keywords and glossary

- Contraband files
- Forensic integrity of digital evidence
- artifacts = pieces of digital evidence that are of importance to the case
- enhanced previewing - better than triage (full drive)
- Indecent Image of Children
- early look intelligence gathering
- Immediate feedback loop for onsite investigators
- "Suspect's dwelling"
- securing a conviction of the offender
- protecting future victims
- browser artifacts
- onsite == in situ
- adhere to proven forensic principles

9 Graphs, Images, Figures, and tables

- graph - aquisition time vs full

- graph - investigation time vs full
- graph - analysis time vs full
- graph - total time from initial plug-in to decision to be evidence (threshold)
- graph - current backlock in ventura county
- figure - DFT vs TCU (Hitchcock[5])
- graph - collection time vs number and size of files
- graph - speed of collecting evidence from same drive based on microSD card used