2015

Open Forensic Devices

Lee Tobin
DigitalFIRE

Pavel Gladyshev
DigitalFIRE

Follow this and additional works at: https://commons.erau.edu/jdfsl

Part of the Computer Engineering Commons, Computer Law Commons, Electrical and Computer Engineering Commons, Forensic Science and Technology Commons, and the Information Security Commons

Recommended Citation
DOI: https://doi.org/10.15394/jdfsl.2015.1214
Available at: https://commons.erau.edu/jdfsl/vol10/iss4/8

This Article is brought to you for free and open access by the Journals at Scholarly Commons. It has been accepted for inclusion in Journal of Digital Forensics, Security and Law by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

(c)ADFSL
OPEN FORENSIC DEVICES
Lee Tobin, Pavel Gladyshev
DigitalFIRE

ABSTRACT
Cybercrime has been a growing concern for the past two decades. What used to be the responsibility of specialist national police has become routine work for regional and district police. Unfortunately, funding for law enforcement agencies is not growing as fast as the amount of digital evidence.

In this paper, we present a forensic platform that is tailored for cost effectiveness, extensibility, and ease of use. The software for this platform is open source and can be deployed on practically all commercially available hardware devices such as standard desktop motherboards or embedded systems such as Raspberry Pi and Gizmosphere’s Gizmo board. A novel user interface was designed and implemented, based on Morphological Analysis.

Keywords: forensic device, open source, write-blocker, forensic imaging, morphological analysis, user interface design

1. INTRODUCTION
The process of acquiring images of hard disks is a well documented and usually straightforward task for a forensic investigator. Still, taking forensically sound images of hard disks is an essential step during investigations involving computer systems.

This paper details a system that provides the same functionality as current forensic hardware at a fraction of the cost. While this platform is presented as a write blocker and imaging device, it is designed to be extended and enhanced. Several example developments are detailed in this paper. This project is an attempt to facilitate the building of forensic devices and to enable forensic investigators to add their own functionality and personalisations.

The novel user interface for this platform is based on ideas from Morphological Analysis. This responsive web user interface provides a clean and informed way of presenting forensic tasks to a user.

1.1 Commercial Forensic Devices
Based on our research, current forensic platforms are expensive, some costing thousands of Euro. While this cost is perfectly reasonable for some organisations and departments, it can be an issue for others.

1.2 FIREBrick System
The FIREBrick Project (Tobin & Gladyshev, 2013) is an open embedded system based on Linux that, in its most basic form, provides write-blocking functionality and provides forensically sound copies of hard disks. The system can be built using almost any motherboard.

The FIREBrick operating system is a
minimal customised Linux distribution. As this device is Linux based, developers can easily develop and enhance the platform using the wide array of available Linux software.

To date, the FIREBrick Project has received interest from the computer forensic community. Several investigators and many students from around the world have developed new functionality for the platform. In particular, a number of new features were developed by M.Sc. students from the Digital Investigation and Forensic Computing programme in University College Dublin.

2. HARDWARE

The FIREBrick is constructed using common, cost-effective, commercially available hardware. Configurations of Raspberry Pi, Gizmosphere’s Gizmo have been built. The desktop computer based device requires the following hardware:

- Motherboard (any form factor)
- RAM
- Case and power supply

2.1 Customised Forensic Device

As the system is open source, it can be tailored for a specific requirement or specific law enforcement agency. Smaller, less powerful hardware could be used and powered by a battery, or a more computationally powerful system could be built that would be larger and less portable.

2.2 Stand Alone Device

To demonstrate the configurable nature of the platform, the FIREBrick can be configured as a stand alone device, not requiring a host system to operate. This configuration might be suitable for investigators that do not wish to carry around a lot of equipment. An LCD screen can be connected to the FIREBrick and a simple menu presented to the user (Tobin, 2013a). This mini-ITX device is shown in Figure 1.

Figure 1. LCD screen FIREBrick device.

Going further with this minimalist concept, a FIREBrick can be configured to automatically begin imaging a disk as soon as the disk is connected to the system, providing an audio alert from the speaker on the motherboard when imaging is complete. An example of this configuration is shown in Figure 2.

Figure 2. Minimalistic FIREBrick device.

These examples have been constructed and are currently being used by forensic investigators. The approximate cost of building a minimalist FIREBrick device is shown below:
2.3 Extensible Platform

The FIREBrick is based on Linux, thus any Linux compatible software is available to FIREBrick. Tools such as The Sleuth Kit (Carrier, 2010), Android Debug Bridge (Developers, 2012) and Photorec (Grenier, 2007) have been used with the FIREBrick. One such configuration of the FIREBrick platform added a WiFi device allowing a user to connect via any WiFi compatible device (Tobin, 2013b). Another configuration performed write-blocking over iSCSI, where the user connected to the FIREBrick via an Ethernet cable (Tobin, 2015).

2.4 Mobile Device Acquisition

As the Android Debug Bridge software is available to the FIREBrick, it can be configured to acquire information from mobile devices such as phones and tablets. While mobile data acquisition is feasible and has been built, it requires further development to support a wider range of makes and models of mobile devices.

2.5 Purpose Built Devices

Another configuration of the FIREBrick is a network packet capture and analysis device. The FIREBrick uses TCPDUMP (Jacobson, Leres, & McCanne, 2003) and can be connected to a target network to capture network packets. These packets could be analysed for information gathering, threat detection purposes et cetera.

2.6 Scripting

Scripting is a powerful way of querying and analysing large data sets. In terms of scripting functionality, FIREBrick has been build and tested with Node.js and with Python. Again, the concept with this platform is to allow developers to add their preferred scripting languages to the platform.

3. SOFTWARE

3.1 Operating System

The FIREBrick operating system (FIREBrick OS) is a custom Linux distribution built from the ground up using Buildroot (Korsgaard, 2015). The FIREBrick OS may be written to the motherboard BIOS, configured as a boot disk, configured as a boot USB key, or via PXE booting. It is a lightweight OS, unlike other Linux based forensic distributions, it comes with a very minimal set of software packages and comes without superfluous libraries. This has benefits in terms of speed, maintainability and security.

The FIREBrick OS may be deployed on many hardware platforms. One example is deploying the OS on a desktop motherboard, where the OS is flashed onto the BIOS. This requires no boot device and the system will never boot from evidential drives. Also, this allows the system to run very quickly and boot almost instantly. There are downsides of this approach, such as, the OS size is limited to the size of the BIOS which is usually relatively small (typically 32Mb). However, a minimal system with write-blocking and imaging capability can easily fit into the BIOS of a desktop motherboard.

3.2 Write Blocking Functionality

Figure 3 shows a typical configuration of a FIREBrick system. The user views and interacts with the target device via an iSCSI initiator, this iSCSI target is read-only and hence the evidential drive cannot be altered in any way. Utilising iSCSI provides remote
access to the FIREBrick device. From the prospective of the FIREBrick OS, the target drive is never mounted and data is never written to the drive. The user can also control the FIREBrick using the web user interface (WUI). The write-blocking functionality is iSCSI in this case, however the system can be configured to write-block via a Firewire or via USB with supplemental hardware.

![Laptop FIREBrick WUI Target](image)

Figure 3. Typical FIREBrick Deployment.

Imaging utilises the DCFLDD (Harbour, 2006) package, a modified version of GNU dd, used frequently by forensic investigators.

### 3.3 Morphological Analysis

Morphological Analysis (MA) is a powerful problem-structuring and problem-solving technique created by Fritz Zwicky (Zwicky, 1948). MA works by analysing a problem space, arranging it into tabular form and identifying inconsistencies. Ritchey extended MA (Ritchey, 1998) using software to aid in the process.

### 3.4 Morphological Analysis User Interface (MAUI)

This novel user interface, based on MA, was designed to facilitate ease of use and provide a way of guiding the user through tasks. The user interface for one configuration of a FIREBrick can be seen to provide an intuitive and helpful way of performing forensic tasks that is quite different to regular menu-driven user interfaces. The web user interface is shown in Figure 4.

![Task DATA SOURCE DATA LOCATION SYSTEMSTATE SYSTEMIMPACT](image)

Figure 4. The MAUI interface.

This interface provides a way of presenting forensics tasks, of grading them, and of accessing each task in an uncluttered fashion. To explain how MA is used as a user interface, we shall consider a simple example. A SATA disk taken from a computer is to be imaged. Figure 5 shows the interface after a user selected Capture from Task.

![Task DATA SOURCE DATA LOCATION SYSTEMSTATE SYSTEMIMPACT](image)

Figure 5. MAUI after selecting Capture from Task.

We can see here that the value of Saved Image from Data Source is unavailable. This is the case because the two values are inconsistent. Figure 6 shows each parameter compared with every other parameter. This facilitates the identification of inconsistent pairs of values.

![Identified inconsistent forensic tasks](image)

Figure 6. Identified inconsistent forensic tasks.
This information would not be available to the user as it is hard-coded into the interface and represents the logic driving the interface. We can see that Capture and Saved Image are identified as inconsistent (highlighted in Figure 6). This is so because a saved image cannot be captured, a saved image is already captured. Similarly, View and Off are inconsistent. A system cannot be viewed that has been turned off for a reasonable amount of time (Halderman et al., 2009; Müller & Spreitzenbarth, 2013).

At this point, Computer from Data Source, Disk from Data Location and Off from System State are selected. Now the impact on the system would be shown. Once None in System Impact is selected, the task is initiated. The penultimate step can be seen in Figure 7. Parameter values can be chosen in any order. However, all parameters must have a value chosen in order to initiate a task. Once all values are selected, a confirmation dialog box will be displayed.

Figure 7. SATA disk imaging task.

Using MA as an interface presents forensic tasks to the user, but also provides more information about the tasks to the user. In our example, it allows the user to see the level of system impact the selected task has on the target system. In this example we saw that there was an impact of None, as the disk was taken from the computer and nothing will be written to the disk.

MAUI is arranged in a way to show the full spectrum of tasks available to the user in an uncluttered and transparent way. There are no nested menus and no extra options as all available tasks are presented to the user. This compact interface is also suitable for mobile devices that have limited screen sizes.

4. RESULTS

In this paper we have detailed a forensic platform that provides the same functionality as commercial forensic devices, at a fraction of the cost. This platform is presented as both a hardware and software platform, the software is similar in some ways to forensic Linux distributions such as SANS Investigative Forensic Toolkit or CAINE, however the FIREBrick OS is built from the ground up. The software only contains what is absolutely necessary to provide forensic functionality. As such, the FIREBrick software is well suited for use with embedded hardware.

4.1 Performance

As the choice of hardware for the FIREBrick device is open-ended, performance can vary quite substantially. For the FIREBrick device in Figure 1, speeds of 5GB/min were obtained while imaging a Kingston SSDNOW SERIES V100S2D SSD drive. Throughput would increase for faster SSD drives and would decrease for slower platter-based drives. The specifications for commercial forensic imaging devices detail speeds ranging from 4GB/min(CPRTools, 2015) up to 15GB/min(Tableau, 2015). These figures have not been verified by the authors of this paper.

4.2 User Interface

The MAUI user interface designed for this project is based on principles from Morphological Analysis. This novel user interface provides a clean and informed way of presenting forensic tasks to a user.
4.3 Extensibility

One of the overall goals of the FIREBrick project is to encourage developers to extend the functionality of the platform. Areas of computer forensics such as mobile device acquisition, computer network analysis and file carving could benefit from having an open, stable, extensible device.
REFERENCES


