THE JOURNAL OF DIGITAL FORENSICS, SECURITY AND LAW

Journal of Digital Forensics, Security and Law

Volume 11 | Number 1 Article 3

2016

Low Budget Forensic Drive Imaging Using Arm Based Single Board Computers

Eric Olson
Sam Houston State University

Narasimha Shashidhar
Sam Houston State University

LOW BUDGET FORENSIC DRIVE IMAGING USING ARM-BASED SINGLE BOARD COMPUTERS

Eric Olson
Department of Computer Science
Sam Houston State University
Huntsville, Texas
ero003@shsu.edu

Narasimha Shashidhar
Department of Computer Science
Sam Houston State University
Huntsville, Texas
karpoor@shsu.edu

ABSTRACT

Traditional forensic analysis of hard disks and external media typically involves a “dead analysis” of a powered down machine. Forensic acquisition of hard drives and external media has traditionally been accomplished by one of several means: standalone forensic duplicator; using a hardware write-blocker or dock attached to a laptop, computer, workstation, etc.; forensic operating systems that live boot from a USB, CD/DVD or virtual machines with preinstalled operating systems. Standalone forensics acquisition and imaging devices generally cost thousands of dollars. In this paper, we propose the use of single board computers as forensic imaging devices. Single board computers can provide a low budget forensic imaging solution that can be used in a lab, remote acquisition, or even be configured as portable imaging devices. This project tests different ARM processor based single board computers and the software available at the present time. The project includes image acquisition using a write-blocker, software write-blockers and without write-blockers to test the various configurations. The final results demonstrate clearly that ARM-based single board computers can serve as effective, low cost and low energy forensic imaging devices.

Keywords: Forensic Imaging; Forensic Acquisition; Single Board Computers; Low Budget Digital Forensics

1. INTRODUCTION

Traditional forensic analysis of hard disks and external media typically involves a powered down machine and performing a “dead analysis” of these storage devices. Forensic acquisition of hard drives and external media can be done by one of several means: standalone forensic duplicators; using a hardware or software write-blocker or a dock attached to a forensic workstation; forensic live boot operating systems; or forensic operating systems installed as virtual machines. Here are some of the issues that we have identified with the current forensic acquisition techniques:

- Standalone portable forensic duplicators (Tableau, ImageMaSSter, Wiebetech, etc.) are prohibitively expensive (several thousands of dollars) for small police departments and similar investigative bodies.
- Using a hardware write blocker or dock attached to a laptop or workstation ties up existing laptop or workstation resources.
- Hardware write blockers generally cost hundreds of dollars, and introduce the possibility of machines locking up while working on other processes.
- Virtual machines with preinstalled operating systems tie-up existing resources and creates the possibility of machines locking up while working on other processes and furthermore, the host operating systems may have compatibility issues connecting USB devices to virtual machines.
- Forensic operating systems that live boot from a USB, CD/DVD on a computer tie-up the existing workstation or laptop and no other processes can be run unless there is another machine available.
- Lastly, if used on the suspect's computer there is the possibility of contamination or the suspect's machine may have some form of anti-forensic tool installed which needs to be dealt with.

These are some of our motivating reasons to study the framework of standalone, low-budget forensic imaging solutions. Current standalone forensics acquisition and imaging devices generally cost thousands of dollars and are prohibitive for small police departments and similar investigative entities. Hence, there is a clear need for low budget forensic imaging devices as well as portable options. As single board computers (SBC) decline in cost and increase in computing power, their potential use for real world applications in forensic imaging become more viable. According to Webopedia (Webopedia, 2016), "A single-board computer (SBC) is any complete computer that is built on a single circuit board and contains functional computer components including the microprocessor, input/output (I/O) and memory".

In this research project, we propose the use of ARM processor based single board computers as forensic standalone imaging devices, remote imaging devices and provide portable solutions using this framework. A forensic technician could use a modified single board computer as a portable imaging device or create multiple imaging devices to use while working on other tasks on their main forensics workstation.

ARM based processors are further explained by Sims (Sims, 2014): "At the highest level, the first difference between an ARM CPU and an Intel CPU is that the former is RISC (Reduced Instruction Set Computing) and the latter is CISC (Complex Instruction Set Computing) ...RISC instructions sets are smaller.... This means that the instruction decoder (the bit that works out what the CPU actually needs to do) is much simpler on a RISC processor, and simpler means less power and greater efficiency." Each of the SBC devices used in this research is ARM-based. ARM based single board computers consume much less energy than a desktop, workstation or laptop.

These devices also have a small footprint and are about the size of a credit card in length and width. Many of these SBC devices can be placed on specially designed racks or cases while reducing space used by larger machines. We explore multiple avenues in this paper regarding single board computers as imaging devices in law enforcement, education and business.

2. BACKGROUND

There have been other open-source projects that have worked to build devices and software for low-budget forensic options. Polstra (Polstra, 2016) offers a potential solution to using single board computers as forensic imaging devices using "The Deck" Linux distribution specifically for Beaglebone single board computers. The primary focus of The Deck and related projects appear to be for hacking or penetration testing. Polstra mentions forensic acquisition using the Beaglebone and udev rules for write-blocking. However, Polstra does not appear to provide any hashed results involving his process but offers an explanation in his book titled, "Hacking and Penetration Testing with Low Power Devices" (Polstra P. , 2014). The scripts created are called 4Deck and the process uses a USB hub as a write-blocker. There does not appear to be any verification at this time that the process seems to be forensically sound.

Champlain College (LCDI, 2015) mentions on their Computer & Digital Forensics blog that they were testing a Raspberry Pi as a mobile forensic imaging device. They claim that their team is testing several different Raspberry Pi imaging configurations for performance, efficiency, and ease of use. The project appears to have slowed down at the end of 2015 and has not received any further updates after several issues were encountered, including hash verification. However, a whitepaper (Champlain College LCDI, 2016) dated February 2016 was released to the website in April 2016. The team has claimed that they have finished their testing of the Raspberry Pi 2 and stated that they ran into a number of difficulties through the project that resulted in them not being able to produce any concrete results. It is interesting to note that the team chose dcfldd over dc3dd due to issues they had with dc3dd. However, we would like to note that we have not faced these hash verification issues encountered by the Champlain College team.

The "Firebrick" project (Gladyshev, 2015) appears to have started sometime in 2013 and a paper was released in late 2015. The Firebrick is a custom Linux OS with write-blocking capabilities built in along with a variety of other forensic functions. The project states that it has been used on a variety of devices, including the original Raspberry Pi, and works well with mini-ITX boards. However, after researching the project on GitHub (Tobin, 2013) many portions of the software have not been updated in three years. There did not appear to be any current builds for SBC devices at this time and the Firebrick has primarily been tested with a specific build of mini-ITX components. The Firebrick components are stated to cost approximately USD $199 and achieved imaging speeds of up to 5 GB per minute.

Unfortunately, many open source projects tend to lose support or are infrequently updated. In our research, we have specifically tested two different ARM processor based single-board computers and the software available at the current time (April 2016). The purpose of our research is not to create a new operating system or hardware for acquiring images. Our project uses hardware and software that is widely supported by the open source community. We use Linux based operating systems including Kali Linux and Sumuri Paladin which have commercial backing with frequent updates. Our research includes image acquisition using a hardware write-blocker, software write-blockers and also without write-blockers to test the various configurations. The final results of the tests performed also add to existing research in the area of forensic acquisition without write blocking (Carlton, 2014) by obtaining images without write-blockers.

3. HARDWARE AND ACCESSORIES

3.1 Test Work Station

The laptop used for testing was configured as follows: Intel(R) Core™ i5-4200U CPU @ 1.60 GHz 2.30 GHz, 6 GB RAM, 500 GB SATA II hard drive, Windows 10 (64-bit), 1 x USB 3.0 and 2 x USB 2.0 ports

3.2 Single Board Computers

There are a variety of single board computers and the market continues to expand. This project required the single board computer to have at least 2 USB ports, an Ethernet port, video output, low power input and the board, including components (w/o display) must be under $100.00 (our definition of low-budget). Any devices that would take too long to deliver or were out of stock or Kickstarter projects at the time were not considered for this project due to viability concerns. Due to time constraints and budgetary reasons, we chose the Raspberry Pi 2 (Foundation, 2016) and Odroid XU4 (Hardkernel, Odroid-XU4, 2015) for testing based on their specifications. These two boards were used for full testing of our image acquisition techniques.

There are other devices which could technically be considered single board computers and low-cost options. However, the Raspberry Pi 2 and Hardkernel Odroid XU4 were chosen due to the community support, features and price. The Raspberry Pi 2 is widely supported for educational purposes, used throughout the maker movement community and has a monthly magazine devoted to Pi devices.

However, the Raspberry Pi 2 is limited to USB 2.0 ports while the Hardkernel Odroid XU4 has two high speed USB 3.0 ports for twice the price of the Raspberry Pi 2. There did not appear to be any other ARM based devices at this time that included USB 3.0 and specifications of the Odroid XU4 for under USD $100. These two SBCs were also chosen due to their availability and ease of purchase. We were unable to find any x86 based SBCs in the same low price range as the ARM-based devices.

After some preliminary tests and analysis of the results, we also purchased two additional single board computers including the Banana Pro (Lemaker, 2016) and Odroid C2 (Hardkernel Odroid C2, 2016). We used these two SBC devices to compare acquisition speeds from an attached media source over a gigabit network connection.

Table 1
Single Board Computers Used for Testing

Full Testing Process		**Included in Acquisition Speeds Over Network**	
Raspberry Pi 2 ($29)	**Odroid XU4 ($74)**	**Banana Pro ($40)**	**Odroid C2 ($40)**
900MHz quad-core ARM® Cortex-A7 1GB RAM 4 x USB 2.0 ports HDMI display port Ethernet 10/100 Micro SD card 5V @ 2A MicroUSB	Samsung Exynos5422 Cortex™-A15 2Ghz, Cortex™-A7 Octa core 2Gbyte LPDDR3 RAM 2 x USB 3.0 / 1 x USB 2.0 MicroSD card Gigabit Ethernet port HDMI display port 5V @ 4A MicroUSB	ARM® Cortex™-A7 Dual-Core 1GB DDR3 SDRAM SATA 2.0 / 2 x USB 2.0 MicroSD card Gigabit Ethernet port HDMI display port 5V @ 2A MicroUSB	Amlogic ARM® Cortex®- A53(ARMv8) 2Ghz quad core CPUs (64 bit) 2GB DDR3 SDRAM 4 x USB 2.0 ports Gigabit Ethernet port HDMI display port MicroSD or eMMC flash 5V @ 2A MicroUSB

3.3 Accessories

- CoolGear USB 3.0 Hardware Write Blocker with IDE/SATA adapters, hardware write block switches and external power supply
- Vantec NexStar USB 3.0 with IDE/SATA adapters, and external power supply
- Sabrent 4 port USB 3.0 hub with individual switches for USB ports
- Patriot 16GB SDHC UHS-I microSD cards
- Samsung Pro 32GB SDHC UHS-I (3) microSD card
- Samsung Evo+ 32GB UHS-I microSD card
- Sandisk Extreme 32GB SDHC UHS-I (3) microSD
- Patriot 8 GB USB 3.0 flash drives
- 3.4 Hard Drives
- Several of the drives available in our laboratory were used for the testing and analysis:
- 20 GB 2.5” IDE Toshiba (failed with a head crash)

- 250 GB 3.5” IDE Maxtor (failed with a head crash)
- 60 GB 2.5” IDE Toshiba (used for testing)
- 160 GB 2.5” SATA II Western Digital (used for testing)
- 120 GB 2.5” SATA III SSD SanDisk (used for testing)

The hard drives below were used for testing and storage purposes.

- 1 TB 2.5” USB 3.0 Western Digital
- 2 TB 2.5” USB 3.0 Seagate
- 5 TB 3.5” USB 3.0 Seagate

4. TESTING FRAMEWORK

The testing framework was revised and closely follows the guidelines stated in (Carlton, 2014) with additional acquisition formats while including the ARM-based devices. As outlined in (Carlton, 2014) the testing process includes these steps with modifications.

First, known data sets are prepared and written to verified wiped storage devices. After the data sets are prepared, the files are hashed and the disk is hashed using Sumuri Paladin. All hardware write-blocked and software write-blocked images were acquired first through Paladin with verified hashes.

Each acquisition was first performed in Paladin (USB boot), followed by Windows 10 with FTK Imager. The process is then repeated using the Odroid XU4 and the Raspberry Pi 2 using dc3dd and ewfacquire. The drives that are acquired are attached via USB cables and the following process is used:

1. Take a bit-stream image (dd) and expert witness format (e01) image of the storage device attached to the device with a hardware write blocker. The hardware write blocker is the CoolGear model that attaches via USB 3.0 cable and with the write block switch on.
2. Take a bit-stream image (dd) and expert witness format (e01) image of the storage device attached to the forensic workstation with a software write blocker. To further verify software write blocking a Vantec Nexstar SATA/IDE adapter is used that attaches via a USB 3.0 cable. This also allows for multiple devices to be run at once using hardware or software write blocking
3. Take a bit-stream image (dd) and expert witness format (e01) image of the storage device attached to the device without using a hardware or software write blocker. If the hash of the image changed from the original verified result the original image was loaded back onto the source drive for use on the next device and verified.
4. Compare disk image results and acquisition speed/time findings.

At the end of testing all raw images are verified using FTK Imager and e01 files further verified using Encase Imager.

5. DEVICE AND SOFTWARE SETUP

Sumuri Paladin is not available for ARM based devices as it is only available for 32 or 64 bit x86 devices. Windows only appears to be available as a limited operating system such as Windows IoT for the internet of things and is not a full Windows OS at this time. There are a multitude of Linux distributions available but Kali Linux was used on the Hardkernel Odroid XU4 and Raspberry Pi 2 due to the ease of installing the forensic meta-packages.

5.1 Sumuri Paladin (Version 6.08)

Sumuri Paladin (Sumuri, 2016) was chosen as the initial forensic acquisition operating system and hash verification tool. Paladin is considered a forensically sound operating system that can be booted into via a USB flash drive and is often used by law enforcement. It is available for free, or by donation, and can be downloaded from Sumuri's website (Sumuri, 2016). Paladin was installed to a USB flash drive using Universal USB Installer on Windows 10. The laptop was then booted from the USB drive with Paladin installed and forensics mode was chosen. Paladin has a utility labeled "Paladin Toolbox" which can be used for a variety of forensic tasks including imaging. Paladin Toolbox was used for this research and has the ability to output two different image formats at once.

5.2 Windows 10 & FTK Imager (Version 3.4.2.2)

FTK imager (AccessData, 2016) and Encase Forensic Imager (Guidance Software, 2016) are considered forensically sound and have been used widely by law enforcement as well as tested by NIST. This project tests the images acquired and compares the hashes to test the forensic capabilities of the implemented software and hardware. Only FTK Imager is used as the free version of Encase imager does not include options for raw bit-stream images. FTK Imager is free, downloaded from AccessData's website (AccessData, 2016) and installed in Microsoft Windows by running the executable. FTK Imager outputs one image at a time and the process is repeated using raw (dd) and e01 formats as output.

5.3 Kali Linux (Version 2016.1 / 2.1.2)

The Odroid XU4 and Raspberry Pi 2 Kali Linux images were downloaded from Offensive Security's website (Offensive Security, 2016) and then unzipped. The SD Cards were formatted as FAT32 using SDFormatter and the images were loaded to the SD card using Win32DiskImager. This process could have also used dd/dc3dd to push the images to the SD card in Linux.

During the first boot of Kali Linux an error regarding file permissions was encountered on the Odroid XU4 and later on during the boot process of the Raspberry Pi 2. The devices refused to boot after this error and we fixed this with the following process:

- On the Raspberry Pi the file is called commandline.txt and on the Odroid XU4 it is called boot.ini. After the following line "root=/dev/mmcblk0p2 rootwait" there was a "ro" for read-only. This was changed to "rw" for read-write and the systems booted properly (odroid XU4 with Kali 2, read only file system, 2015).

After the first boot the following commands were issued through a terminal in Kali Linux on the Odroid XU4 and Raspberry Pi 2 (Kali Linux Official Documentation - Kali Linux - Raspberry Pi, n.d.).

The default password and SSH keys were changed using the following commands (default user is root and default password is toor)

- sudo passwd
- rm /etc/ssh/ssh_host_*
- dpkg-reconfigure openssh-server service ssh restart

The packages and software were updated by using the following commands:

- sudo apt-get update && sudo apt-get dist-upgrade

The Kali Linux forensic packages (including dc3dd and ewfacquire) were installed by entering the following:

5. sudo apt-get install kali-linux-forensic.

This process can take some time on the Raspberry Pi 2 and significantly less time on the Odroid XU4.

5.4 Forensic Utilities for SBC devices

All tools listed below can be found on the Forensics Wiki (Forensicswiki - Category: Disk Imaging, 2016):

- DCFLDD – command line Linux
- DD – command line Linux
- DC3DD – command line Linux
- DDRescue – command line Linux
- EWFAcquire – command line Linux tool for imaging to Expert Witness Format (EWF/E01)
- FTK Imager - command line Linux tool does not appear to work for ARM devices at this time
- Guymager – not available for ARM at this time
- AIR (Automated Image & Restore) – not available for ARM at this time
- AFFLIB/AFF4 – does not appear to acquire images from devices at this time but can convert image files using affconvert or affcat tools
- Aimage – image acquisition for AFFLIB but support has been dropped and it does not appear to work on ARM devices.

At this time, we were unable to find a technique to acquire images in the AFF format on ARM devices. The Linux based forensic utilities used in this research are dc3dd and ewfacquire. DCFLDD was not used due to a potential issue explained in a NIST test (NIST & US Department of Homeland Security, 2013) (12/2013) and DCFLDD does not appear to have been updated since 2006. DC3DD is also listed as a forensic imaging tool on the Department of Defense Cyber Crime Center's website (DC3 / Defense Cyber Crime Institute (DCCI), 2016) but DCFLDD is not listed on the DC3 website.

The command line tool help pages can be accessed by typing dc3dd --help or ewfacquire --help. The command line syntax used for ewfacquire and dc3dd are shown below:

- fdisk -l and lsblk commands were used to find information about the attached drives
- To make an image of a hard drive, USB drive, etc. with dc3dd:
  - sudo dc3dd if=/dev/[device such as sda] of=/[location/filename] hash=md5 hash=sha1 log=[logname].log [bufsz=xx]
- To verify the hashes of the acquired image with dc3dd:
  - sudo dc3dd if=/[filename] hash=md5 hash=sha1 of=/dev/null log=[logname].log
- To make an image of a hard drive, USB Drive, etc. with ewfacquire:
  - ewfacquire /dev/[device such as sda] -l [logname].log -t [filename]
  - Do not have to include -t (target) as ewfacquire will prompt for remaining information
- To verify the hashes of the acquired image with ewfacquire:
  - ewfverify -d sha1 [filename] -l [logname].log

5.5 Powering external drives

The Odroid XU4 is able to power external 2.5 inch hard drives or multiple USB flash drives without issue. The Raspberry Pi 2 needed modification to the config.txt file which is not available on Kali Linux for the Raspberry Pi 2. A valid config.txt file was copied from an existing Raspbian installation. After the config.txt file was copied to the boot partition the line "max_usb_current=0" was changed to "max_usb_current=1". This modification changed the available amperage to the four USB ports to 1.2A compared to the 0.6A default (Raspberry Pi Forums - Any negative impact with setting `max_usb_current=1`?, 2015). The Raspberry Pi 2 was then able to power an external 2.5" drive attached via USB.

5.6 HDD Setup

The project originally tested a 60GB and 250GB HDD's as well as a 20GB HDD. However, the 250GB HDD failed with a head crash after testing the write-blocker while the 20GB HDD failed with a head crash early on in the project. The project continued with three different hard drives including a 2.5'' 60GB IDE HDD, 120GB 2.5" SSD SATA III HDD and a 160GB 2.5" SATA II HDD.

All drives were first wiped with Sumuri Paladin Toolbox's wipe feature. The 60GB drive was formatted as NTFS and a known set of hashed files were copied to the drive. The 120GB SSD drive was formatted as NTFS and a known set of hashed files were copied to the drive. The files were then hashed again using Quickhash to verify there were no changes to the files after they were copied to the drives.

The 160GB drive was modified significantly from the other two drives and was formatted with four different file system partitions including NTFS, FAT32, EXT4 and HFS+. A known set of hashed files were copied to each partition. The files were hashed again using Quickhash (TedTechnology, 2016) to verify there were no changes to the files after they were copied to the drives.

6. HASH RESULTS

6.1 Hardware Write-Blocker Testing

The CoolGear hardware write blocker was first tested by loading random data onto two hard drives and verifying the hashes of the drive in Sumuri Paladin. The two drives were then acquired using FTK Imager on Windows 10 with the hardware write-blocker. The results were confirmed with software write-blocking using FTK Imager in Windows 10. The hardware write blocker was then used to acquire the images of the two drives using the Odroid XU4 and Raspberry Pi 2. The hashes matched across the devices and in software write blocking in Microsoft Windows 10.

Table 2
Hardware Write-Blocker Test Hashes

60GB IDE 2.5"		
RAW	MD5	6724c73173142678ace5eaaa4944b1db
RAW	SHA1	2ba89ebfb1e11c41252ad6dfa4056faedc25c45c
250GB IDE 3.5"		
RAW	MD5	1927da90530d9e10098ba54e395adf0a
RAW	SHA1	f3a78fe3d28251dc6d85f80071ffba06e6577384

The CoolGear write blocker proved effective across all the devices tested. Additional attempts were made to wipe, format, copy files, move files, etc. but the write blocker would display various errors.

6.2 Hardware Write-Blocking

The hardware write blocking results were the same across all devices tested and the hash results are shown below.

Table 3
Hardware Write-Blocking Hashes

60GB IDE 2.5"		
RAW	MD5	f52c0ec005c50eb51eedf549f81e4d15
EWF	MD5	f52c0ec005c50eb51eedf549f81e4d15
RAW	SHA1	86c9a4a596fa550b90ac5d4bfa357b291dd986cf
EWF	SHA1	86c9a4a596fa550b90ac5d4bfa357b291dd986cf
160GB SATA 2.5"		
RAW	MD5	055c04ce6f666c2143cb50eb2d388946
EWF	MD5	055c04ce6f666c2143cb50eb2d388946
RAW	SHA1	6449f2543bd040c4ff113e24a33ec3df16d26aae
EWF	SHA1	6449f2543bd040c4ff113e24a33ec3df16d26aae
120GB SATA SSD 2.5"		
RAW	MD5	290e37ef8c68d6948be9f25c0c5ebfb5
EWF	MD5	290e37ef8c68d6948be9f25c0c5ebfb5
RAW	SHA1	b29a9a8711e652d04461a283e56ad165cda5b579
EWF	SHA1	b29a9a8711e652d04461a283e56ad165cda5b579

6.3 Software Write Blocking

Software write-blocking was obtained by different methods across the devices.

Sumuri Paladin – Paladin Toolbox allows for software write-blocking of attached storage devices when booting from a USB into the Paladin forensic interface.

Microsoft Windows 10 - Software write blocking was achieved modifying the registry setting located in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\StorageDevicePolicies and WriteProtect dword (32 bit) of 1 (Carlton, 2014). The system was then rebooted and the USB software-write block was verified with a USB drive.

Kali Linux on SBC devices – This research utilizes (MSuhanov Linux Write Blocker, 2016) by copying the udev forensic read only rule to the /etc/udev/rules.d folder and the additional scripts (tools) to /usr/sbin. This method blocks mounting of block devices connected to the SBC. The system was then rebooted and the USB software-write block was verified with a USB drive. If a storage device is attached and to be used for storage it must be set writeable by issuing blockdev --setrw /dev/* (where * is device/partition such as sda1).

The software write blocking results were the same across all devices tested and the hash results are shown below.

Table 4
Software Write-Blocking Hashes

60GB IDE 2.5"		
RAW	MD5	f52c0ec005c50eb51eedf549f81e4d15
EWF	MD5	f52c0ec005c50eb51eedf549f81e4d15
RAW	SHA1	86c9a4a596fa550b90ac5d4bfa357b291dd986cf
EWF	SHA1	86c9a4a596fa550b90ac5d4bfa357b291dd986cf
160GB SATA 2.5"		
RAW	MD5	055c04ce6f666c2143cb50eb2d388946
EWF	MD5	055c04ce6f666c2143cb50eb2d388946
RAW	SHA1	6449f2543bd040c4ff113e24a33ec3df16d26aae
EWF	SHA1	6449f2543bd040c4ff113e24a33ec3df16d26aae
120GB SATA SSD 2.5"		
RAW	MD5	290e37ef8c68d6948be9f25c0c5ebfb5
EWF	MD5	290e37ef8c68d6948be9f25c0c5ebfb5
RAW	SHA1	b29a9a8711e652d04461a283e56ad165cda5b579
EWF	SHA1	b29a9a8711e652d04461a283e56ad165cda5b579

6.4 Without Write Blocking

For this part of the testing, we acquired each drive without write blocking. The drive was acquired first in raw format and then in expert witness format. When the drive hash results changed the drive was reimaged with the write-blocked image using Sumuri Paladin. The drive was then verified to have the original hash result of the write-blocked image before moving on to the next device for imaging.

The SBC's were loaded with new Kali Linux installations on different microSD cards. The udev scripts used in the previous software write-blocking step were not used on the SBC's for this test. The hashes of the drives did not change when imaging the drives using Kali Linux on the SBC's and the drives appeared to be mounted as read-only.

Table 5
Without Write-Blocking Hashes

SBC's Kali Linux and Sumuri Paladin Without Write-Blocking		
60GB IDE 2.5"		
RAW	MD5	f52c0ec005c50eb51eedf549f81e4d15
EWF	MD5	f52c0ec005c50eb51eedf549f81e4d15
RAW	SHA1	86c9a4a596fa550b90ac5d4bfa357b291dd986cf
EWF	SHA1	86c9a4a596fa550b90ac5d4bfa357b291dd986cf
160GB SATA 2.5"		
RAW	MD5	055c04ce6f666c2143cb50eb2d388946
EWF	MD5	055c04ce6f666c2143cb50eb2d388946
RAW	SHA1	6449f2543bd040c4ff113e24a33ec3df16d26aae
EWF	SHA1	6449f2543bd040c4ff113e24a33ec3df16d26aae
120GB SATA SSD 2.5"		
RAW	MD5	290e37ef8c68d6948be9f25c0c5ebfb5
EWF	MD5	290e37ef8c68d6948be9f25c0c5ebfb5
RAW	SHA1	b29a9a8711e652d04461a283e56ad165cda5b579
EWF	SHA1	b29a9a8711e652d04461a283e56ad165cda5b579

The only time the hash of the drives appeared to change was when using FTK Imager in Windows 10 without write blocking as shown below in Table 6.

Table 6
Windows 10 - Without Write-Blocker Hashes

Windows 10 FTK Imager 3.4.2.2 Without Write-Blocking		
60GB IDE 2.5"		
RAW	MD5	1b41bc84784434b03c232836171afc4d
EWF	MD5	373782bada1930db65572533cee32f2b
RAW	SHA1	29d4d9d0865edc13267f37e8227222428f5bcf28
EWF	SHA1	8401cfe4c862eb920047a025af4a52f4d74c9a5d
160GB SATA 2.5"		
RAW	MD5	04f14a2c92cc733c7946bf358c873c65
EWF	MD5	04f14a2c92cc733c7946bf358c873c65
RAW	SHA1	b22bd3b742e7fbba891c995f9b135ab65abd2ee7
EWF	SHA1	b22bd3b742e7fbba891c995f9b135ab65abd2ee7
120GB SATA SSD 2.5"		
RAW	MD5	3e5eaab9a63ffc2a41aedacef97be95a
EWF	MD5	dd3310f0d3fa22c0e8c5475a436166b9
RAW	SHA1	cb83f2d7dc13721e7aae419fb2d0ba3b902f0ac0
EWF	SHA1	8aeccbda8935fdff13969d74690876ce0c5daebc

7. IMAGING SPEEDS AND TIMES

7.1 Test speeds of USB devices and SD cards used

There are a variety of factors that can impact the speed and time of an acquisition. The Raspberry Pi 2 is limited to four USB 2.0 ports while the Odroid XU4 has one USB 2.0 port and two USB 3.0 ports. The table below lists the theoretical maximums of common interfaces for reference. However, it is rare to see the theoretically maximums reached due to various influencing factors. The data transfer speeds were calculated using an online converter (Convert-Me, 2016) and are shown in Table 7.

Table 7
Theoretical Transfer Speeds of Common Interfaces

Interface	Gbps	MBps	GB per min	TB per hour
100Mb eth	.1	12.5	.75	.045
USB 2.0	.48	60	3.6	.216
1Gb eth	1	125	7.5	.45
SATA I	1.5	187.5	11.25	.675
SATA II	3	375	22.5	1.35
USB 3.0	5	625	37.5	2.25
SATA III	6	750	45	2.7
USB 3.1	10	1250	75	4.5

Read and write speeds of the USB devices and SD cards used in this research were obtained using DiskMark (NetworkDLS - DiskMark, n.d.) on Windows 10. The hard drives were attached to the USB 3.0 port with the NexStar USB 3.0 adapter and the microSD cards were tested while inserted into a Sabrent USB 3.0 adapter. The results are displayed below in Table 8.

Table 8
Maximum Read/Write of Tested Storage Media

Storage Device	Max Read MB/s	Max Write MB/s
SanDisk Plus 120GB SSD	253.3	213.1
Seagate 5TB 3.5” SATA III	138.3	137.0
Seagate 2TB 2.5” SATA III	109.3	111.2
WD 1TB 2.5” SATA	82.8	81.1
WD 160GB 2.5” SATA	62.5	61.8
Toshiba 60GB 2.5” IDE	30.0	30.0
Patriot 8GB USB 3.0 Flash	150.8	30.1
Patriot 16GB Class 10 U1	84.2	12.5
Samsung Evo+ 32GB U1	91.6	46.6
Samsung Pro 32GB U3	91.6	77.6
Sandisk Extreme 32GB U1	93.8	49.4

7.2 Default Testing Image Acquisition Speed

The default speeds of the single board computers were generally slow due to image acquisition from one USB drive to another attached USB drive. The Raspberry Pi 2 barely passed speeds of 10 MB/s at times and slowed to transfer speeds of 1.5 MB/s. The Odroid XU4 would occasionally reach 20MB/s but would then slow to speeds around 10 MB/s. The Windows laptop was generally 2-3 times faster than the Odroid XU4 and up to 5 times faster than the Raspberry Pi 2 at hard drive image acquisitions of up to 60 MB/s.

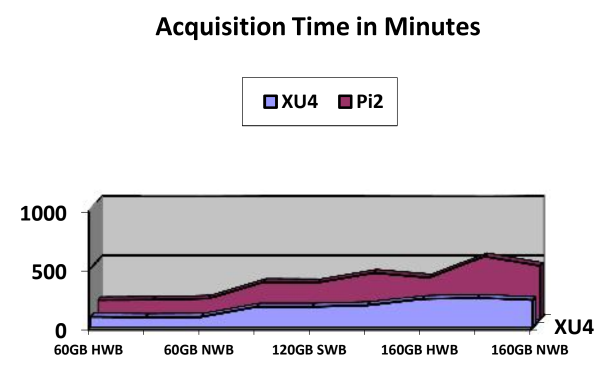


Figure 1. Acquisition Time in Minutes of XU4 and Pi2

Our research shows that the real-world forensic imaging speeds of the single board computers are relatively slow using default settings. The Odroid XU4 allows the option of using an SD card or eMMC card but the eMMC card appears to be a significant price increase over an SD card. Five 16GB UHS-I SD cards can be purchased for approximately the same amount as one 8GB eMMC card. However, higher quality micro SD cards, eMMC modules for Odroid, and SSD drives can significantly increase the acquisition speeds if used to directly acquire the image to the card.

7.3 Increasing Image Acquisition Speeds of SBC's

Additional research was conducted on how to make the image acquisition speeds faster and potential solutions were found. A more thorough test with a USB drive would more effective and quicker in helping us understand the acquisitions speeds. An 8GB Patriot USB 3.0 drive and Samsung 16GB USB drive were used to test acquisition speeds across the devices. The 8GB USB 3.0 drive had maximum read speeds of 150.8 MB/s and write speeds of 30.1 MB/s as indicated by the testing in section VI, A.

We also used faster SD cards including: Samsung Pro 32GB in the Odroid XU4 and the Samsung Evo+ 32GB in the Raspberry Pi 2. We tried three different methods of acquiring an image of the USB drive including: directly to the SD card, to the attached Sandisk 120GB SSD drive formatted as EXT4 and to the attached Sandisk 120GB SSD drive formatted as EXT4 with the root filesystem running from the SSD drive. Ewfacquire was not used at this time as it was unable to acquire the 8GB USB drive due to a known issue with certain USB drives [25]. The following tests involved using different buffer sizes in dc3dd by changing the bufsz= option. Different buffer sizes appeared to increase acquisitions speeds. The chart below shows the time in seconds each acquisition took using different buffer sizes. The Laptop times are shown for comparison. The test times in seconds did not include verification of the image. The image was hashed and compared with the other images using FTK Imager. The hash of the 8GB flash drive did not change after all of the forensic imaging tests using dc3dd. We can see in Figure 2 that the Laptop was 1.5 -2 times faster than the Odroid XU4 and 4 - 5 times faster than the Raspberry Pi 2.

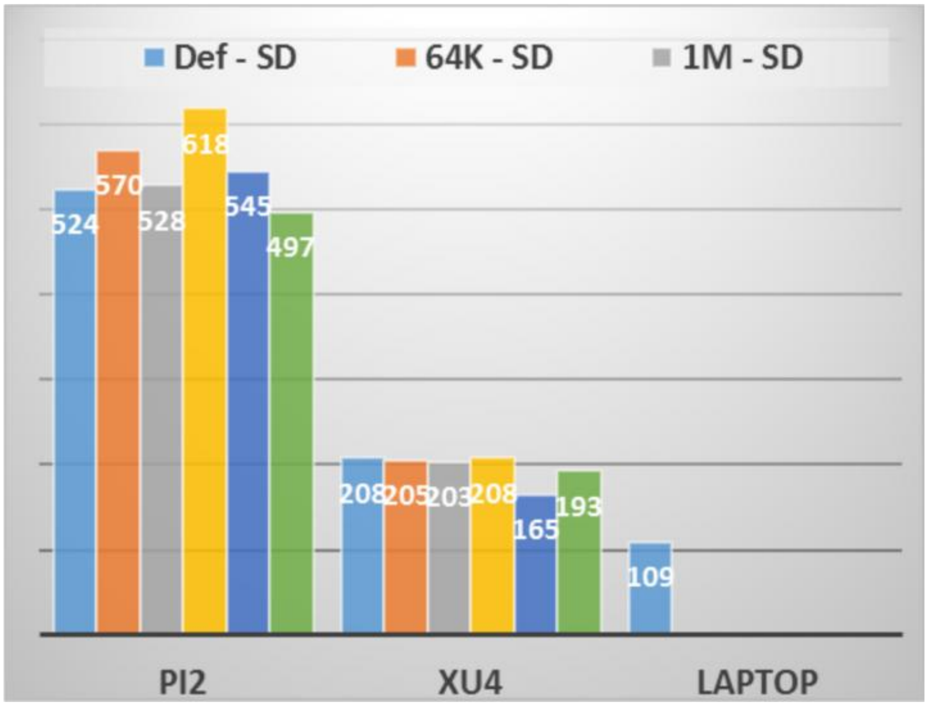


Figure 2. Acquisition Time in Seconds Using USB 3.0 Drive

We can see in Figure 2 above that the Laptop was 1.5 - 2 times faster than the Odroid XU4 and 4 - 5 times faster than the Raspberry Pi 2. The hash of the 8GB flash drive did not change after all of the forensic imaging tests using dc3dd as shown below in Table 9.

Table 9
USB 3.0 Drive Hashes

8GB USB Flash Drive – Hash did not change		
RAW	MD5	d6ec1cbc01ecda2beb956d3a9dcd997d
RAW	SHA1	eb59fbd1a75fc7e743baed035b550953a65a1b1b

1. SD Card as storage - Acquiring an image directly to the SD card of the Raspberry Pi 2 yielded imaging speeds of under 20 MB/s that would drop approximately 5-10 MB/s. The Odroid XU4 displayed speeds of around 60 MB/s that would fluctuate and then drop in speed by 10-20 MB/s.
2. SSD drive attached for storage - The acquisition speeds remained relatively the same as above using an attached SSD drive to image the USB drives to. The Odroid XU4 imaging speeds reached approximately 60 MB per second with an external SSD SATA III drive attached to one of the USB 3.0 ports. The Raspberry Pi 2 again did not cross 20MB/s imaging from the USB flash drive to the attached SSD drive. The speeds did not fluctuate much using this method compared to acquisition straight to the SD card. However, Raspberry Pi 2 would settle at about 14 MB/s while the Odroid XU4 would settle between 40-55 MB/s.
3. SSD drive attached and running root file system - The Odroid XU4 and Raspberry Pi 2 were unable to boot

directly from a USB drive and need the boot partition (/dev/mmcblk0p1) on the SD or eMMC card. However, moving the root filesystem (rootfs on /dev/mmcblk0p2) of the SBC's Linux operating system to an externally attached USB SSD drive increased image acquisition speeds up to approximately 20 MB/s for the Raspberry Pi 2 and up to approximately 60 MB/s per second for the Odroid XU4. However, the speeds would again decrease as the acquisition progressed.

a. Raspberry Pi 2
   *Attaching an external SSD drive and copying the root file system from the SD card to the SSD drive. This can be done a few different ways but this project used the method below based on Bearnes' tutorial (*Bearnes, 2015). *If the UUID method does not work substitute UUID with /dev/sda1 as this is the partition that will first load as the root file system.*
   - Formatted the SSD drive as EXT4
   - Rsync -ax root directory of / from /dev/mmcblk0p2 to the new formatted root directory on the SSD drive
   - In terminal ran blkid to obtain UUID of /dev/sda1
   - Modified the /boot/cmdline.txt file from root=/dev/mmcblk0p2 rootfstype=ext4 to root=UUID=**** rootdelay=5
   - Modified the /etc/fstab and commented (#) out the rootfs associated with the /dev/mmcblk0p2 but added the new partition as the root file system at the bottom as /dev/disk/by-uuid/**** / ext4 defaults,noatime 0 1

b. Odroid XU4
   The process is relatively the same for the Odroid XU4 as shown above with the Raspberry Pi 2. However, instead of changing the cmdline.txt file, the boot.ini file was modified on the Odroid XU4.
   - Formatted the SSD drive as EXT4
   - Rsync -ax root directory / of the sd card from /dev/mmcblk0p2 to the new formatted root directory on the SSD drive
   - In terminal ran blkid to obtain UUID of /dev/sda1
   - Modified the /boot/cmdline.txt file from root=/dev/mmcblk0p2 to root=UUID=**** rootdelay=5
   - Modified the /etc/fstab and commented (#) out the rootfs associated with the /dev/mmcblk0p2 but added the new partition as the root file system at the bottom as /dev/disk/by-uuid/**** / ext4 defaults,noatime 0 1

Using the above method, the devices appeared noticeably quicker but the acquisition speeds did not change from the previous methods. However, it is easier to have the device run from an SD card and use the entire storage capacity of the attached hard drive to store image files if needed. There were a few occasions where the devices refused to reboot with the SSD drive attached as the root file system. It was sometimes necessary to change the root file system back to the original SD card to boot the device.

8. ADDITIONAL RESULTS

8.1 Image Size and Compression

The research conducted in this paper was limited to raw (dd) format and expert witness (e01) format due to limitations on the ARM based single board computers. At this time, we were unable to find a way to acquire the drives in AFF format on the Raspberry Pi 2 as well as the Odroid XU4. The affcat tool allows for conversion of an image but not acquisition from a hard drive. However, the maximum compression ratios were used in FTK Imager on Windows 10 which yielded the following compression results (Figure 3) for dd and e01.

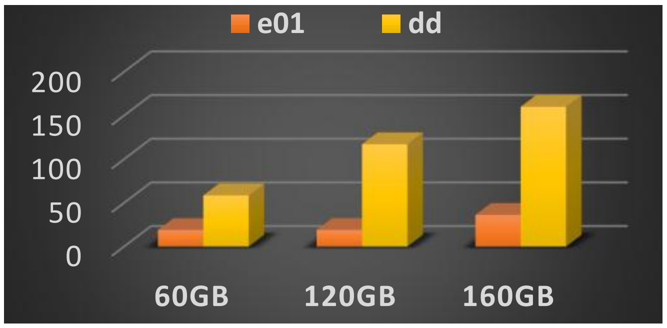


Figure 3. E01 vs. DD Image Size Comparison

The e01 format would be a clear choice in image acquisition using the single board computers with limited space on the device. It may not always be possible to use ewfacquire to acquire an image as shown in the results of the research. However, a raw/dd image can be converted later to e01 or aff format using ewfacquire, affconvert/affcat or other tools.

8.2 Energy Costs

Table 10
US Electricity Summary centers per kilowatt hour

U.S. Electricity Summary cents per kilowatt hour (US Energy Information Administration, 2016)				
	2014	2015	2016	2017
Residential Sector	12.52	12.67	12.58	12.87
Commercial Sector	10.74	10.59	10.59	10.79
Industrial Sector	7.10	6.90	6.91	7.04

The devices were then plugged into a Kill A Watt EZ Power Monitor for twenty four hours to determine power consumption. The Raspberry Pi 2 displayed .03 kWh, the Odroid XU4 displayed .10 kWh and the ASUS laptop displayed .18 kWh over a 24-hour period.

The results below are shown in kWh over 24 hours' x 365 days. The cost per year was calculated by multiplying 365 x (24hr) kWh x 2016 residential cost (Table 10). The maximum kWh was calculated by using the watt of the power adapter or by multiplying voltage x amps to obtain the wattage of the power adapter. The following formula of W $\times$ 24h / 1000 = kWh was used to obtain the maximum power consumption (Figure 4).

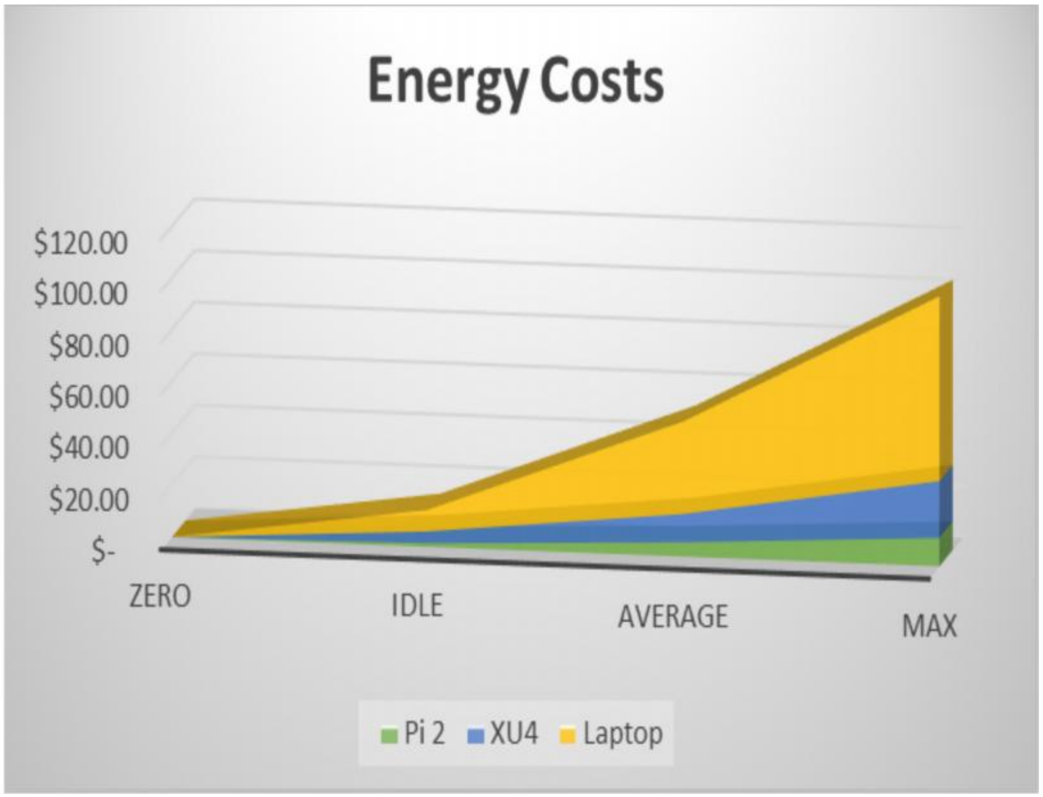


Figure 4. Energy Cost Comparison

The Raspberry Pi 2 uses a power adapter with a maximum of 10 watts, the Odroid XU4's power supply is a maximum of 20 watts and the laptop power supply is a maximum of 65 watts. Keep in mind the cost savings would increase significantly if using a desktop or even a high end forensic workstation with a power supply of over 1000 watts.

9. BUILDING A USABLE MINI LAB

During the research each device was individually attached to a monitor via the HDMI port and a wireless keyboard with touchpad was used for the devices. The process of switching the hard drives to each device, attached the USB dongle for the keyboard, attaching the HDMI cable, and acquiring the image became slow and tedious. This method was used to ensure accuracy of the images that were acquired.

After obtaining the images and verifying the hashes the devices appeared to acquire images that matched across devices. If only using a few single board computers is needed they could be attached to a monitor with two HDMI outputs and with a keyboard. The process of switching back and forth is not too time consuming.

The Raspberry Pi 2 and Odroid XU4 both have compatible touch screen displays to make them a standalone solution or even portable. However, the price may increase to a point at which a low budget laptop around $200 may work as a portable solution. The single board computers are only the size of a credit card in width and length which makes them a space saving solution over a laptop

We looked into cost effective ways to reduce clutter and create a useable lab where several single board computers could be used at once. The process of attaching a screen to each device or using an HDMI switch would increase costs. The most economical choice we found was to create a "headless" option without a monitor using VNC and if needed using SSH for a secure connection.

VNC was used in this process because we wanted command line tools but this also provided the option of using the GUI based tools such as Sleuthkit/Autopsy. This also leaves the option of creating our own GUI based image acquisition tool for ease of use. There are various pieces of open source software that can be used on these devices but x11vnc (Runge, n.d.) was chosen due to its ease of use and ability to view the X interface on display 0. We could then attach a monitor or view the same display through VNC. However, there is no need to attach a keyboard or monitor after initial setup so the use of any VNC software such as TightVNC server would work.

9.1 X11VNC Setup

- Apt-get install x11vnc
- Run x11vnc -display :0 and setup the password and choose yes to storing the password
- Create a startup option by going to applications – session and startup – add
    - Add an option with x11vnc, auto start and command = x11vnc -forever -display :0 -geometry 1280x720
- We then installed a VNC viewer on our Windows laptop and were able to start a VNC session with the "IP address":0
- We also wanted to reboot the ARM devices without having to login or need to attach a monitor. We edited the following:
    - Modify /etc/lightdm/lightdm.conf in two different locations in the file by uncommenting (delete # and add after =)
- autologin-user=enter username
- autologin-user-timeout=0

9.2 Mini Lab and File Server

Each single board computer running Kali Linux is automatically assigned the same hostname of kali. We modified the hostname of each device to reflect the device used, color coded the network cable and changed the background of the desktop to reflect the color of the network cable. This reduced confusion when quickly trying to figure out which device we were using with VNC and what drives were attached where.

Each single board computer can be purchased with a case and there a variety of cases to use or 3D print. However, we wanted an open air type case where we could stack the devices, increase air flow and move them around if needed. We located a stackable solution for attaching the single board computers as a small tower and assembled. However, various other solutions exist and if a 3D printer is available there is the possibility of a completely customized design.

To stay with the theme of using ARM based devices we setup a small file server using an Odroid XU4 and a Cloudshell case (~$40) from (Hardkernel, n.d.). We also used the freely available Linux operating system DietPi (Knight, n.d.) to quickly create a samba file server using the Odroid XU4, Cloudshell and a 120GB SSD drive for testing Mini lab and file server shown below:

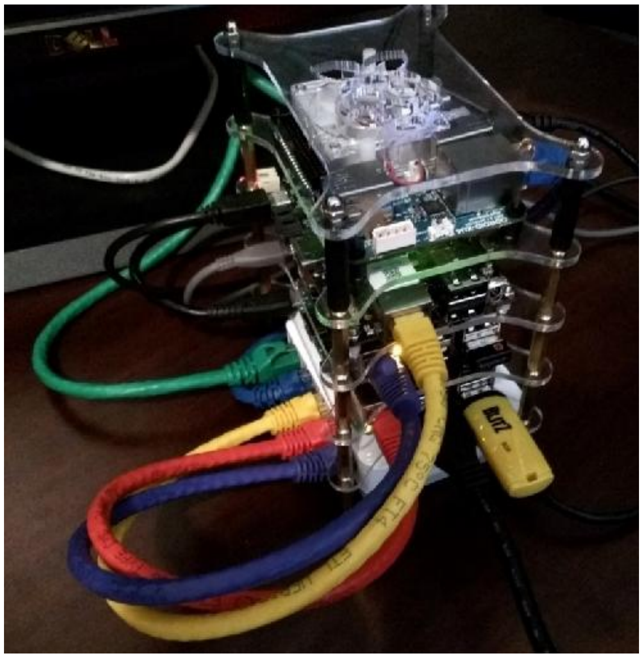
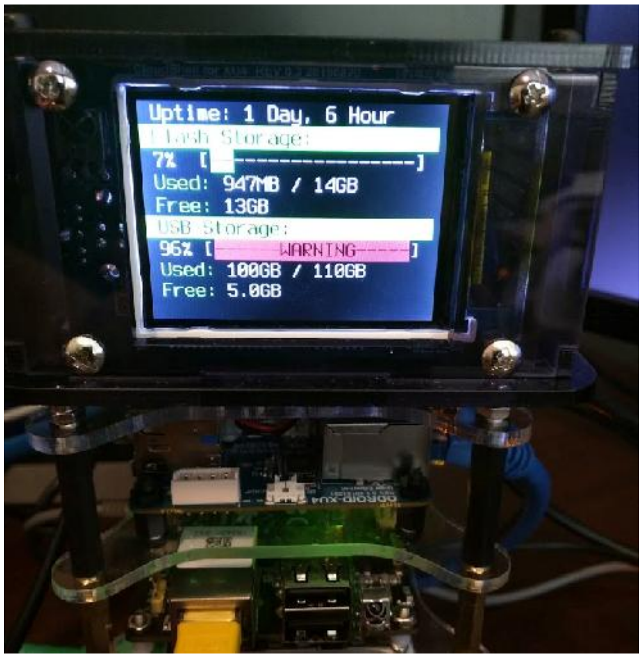


Figure 5. Digital Forensic Mini-Lab



```
Uptime: 1 Day, 6 Hour
Flash Storage:
7% [-----------------]
Used: 947MB / 14GB
Free: 13GB
USB Storage:
96% [------WARNING-----]
Used: 100GB / 110GB
Free: 5.0GB
```

*Figure 6.*Odroid Cloudshell Server Running Dietpi

At this point we had also purchased an Odroid C2 and a Banana Pro to further test image acquisition over a gigabit network using the same Patriot 8GB USB flash drive. Other tests may vary due to network setup but this gives a good idea as to the limitations and real world testing of the devices. The results are shown in Figure 7.

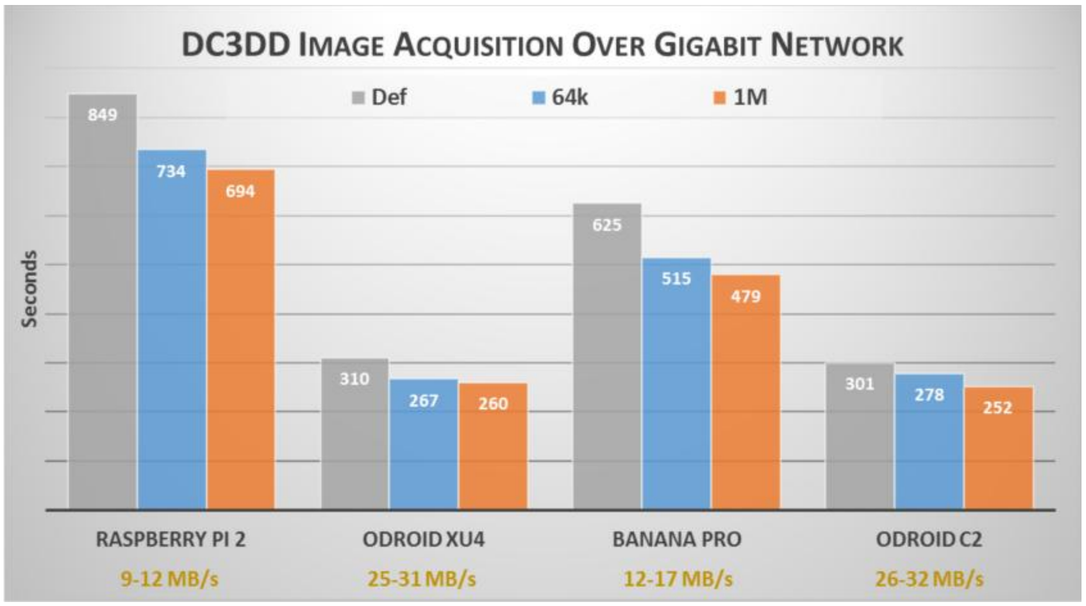


Figure 7. Comparison - Acquisition Speeds Over Gigabit Network

10. FURTHER DISCUSSION AND USE CASES

Several issues were encountered regarding the use of single board computers as imaging devices. However, steps were made and the process was refined. Once the initial hurdles were resolved the process became relatively simple and the SBC's functioned as low cost forensic imaging devices. Although this solution may not work for everyone, the single board computers are highly customizable and accommodate a low budget.

10.1 Low Energy Cost, Small Size and Customizability

The single board computers used in testing consume less energy than typical laptops and significantly less energy than a typical workstation or desktop. These devices can run twenty-four hours a day imaging hard drives at minimal cost per year. The size of a single board computer is also minimal when compared to a laptop or desktop. These devices are designed at about the size of a credit card in length and width. A single board computer can be highly customized as a "headless" solution without a keyboard or monitor; small portable solution including battery source and screen; or as a standalone solution with a small screen and accessories.

10.2 Education

SBC devices appear to be usable as forensic imaging devices and viable options in an educational setting due to their cost. An entire lab of SBC devices could be used to educate students at a fraction of the cost of standard computers, laptops or workstations. They can also assist students who need to purchase another device for digital forensic testing purposes.

The single board computers tested in this research use significantly less energy which reduces overhead in electricity costs that are usually associated with the power consumption of forensic workstations. The single board computers are extremely compact at the approximate length and width of a credit card compared to a larger laptop, desktop or a mini form factor pc. These devices take up little space and some devices such as the Raspberry Pi 2, Odroid C2, and Banana Pro can be powered by a USB hub, multi-port USB charger or even a USB battery pack.

10.3 Digital Forensic Labs

These devices could allow a forensic technician to have multiple imaging devices while working on other tasks on their main forensic workstation. An entire forensic lab of SBC devices could be purchased at a fraction of the cost of standard computers, laptops, workstations or standalone solutions. These devices can be used as standalone forensic imaging stations with a separate monitor or as a headless solution using VNC or other remote connection tools. These devices may assist a digital forensics lab in reducing backlogs of low priority cases. A forensics technician can install one or several of these without taking up much space.

There is also the potential for single board computers to be used as a forensic workstation to conduct a full analysis of an external drive. A report could be generated with the additional software already installed. In brief testing other programs such as Sleuthkit Autopsy, Digital Forensic Framework, LibreOffice and other open source tools could be used.

10.4 Legal Issues and Court Admissibility

Our testing shows that the hashes of the hard drives and USB drive did not change when using software write-blocking methods. Further and independent testing is always required with new or varied digital forensic methods. We believe that with additional testing by other researchers, our methods would be court admissible in legal proceedings. The argument for the validity of open source tools has already been established by various papers including: (Carrier, 2003), (Morra, 2013), and (Garrie, 2014)

11. CONCLUSION

Standalone forensic imaging devices tend to cost thousands of dollars. We found that there were many options when researching standalone forensic imager devices, workstations and low budget laptops. However, low budget options such as laptops, tablets and mini-pc's around the $200 price range were limited. We wanted to test the possibility of using low cost (under $100) single board computers as forensic imaging devices and the process appears to be a success. Additional testing is always needed as the digital forensic testing possibilities are overwhelming.

Four different drives were imaged using the Raspberry Pi 2 and the Odroid XU4. The hash of each drive remained the same through the image acquisition process using hardware write-blocking and software write-blocking. The only time the hash of a drive appeared to change was when using Windows 10 and FTK imager without write-blocking. Even without write-blocking the hash of the drives did not change on the single board computers.

We intentionally did not create a new operating system or piece of hardware. This project used existing community supported resources and hardware. We believe this is better for the digital forensics community as a whole as the resources are already provided. Since Windows is not available for these devices we used Kali Linux and other open source software. Through trial and error, we were able to obtain forensic images from various hard drives and a USB drive in different manners. Further step by step processes can evolve to setup these devices in a forensically sound manner. We have also created a bash script which can install all the necessary software with little user interaction.

Once the device is setup as needed the micro SD card can be imaged and distributed to many other single board computers of the same model. This reduces setup time significantly across devices.

There are many single board computers being released every year that could prove effective as digital forensics solutions. While this project was underway the Raspberry Pi Zero, Raspberry Pi 3, Odroid C2 and various other single board computers were released. The Raspberry Pi Zero is only five dollars and there are many other single board computers available for under $100. These devices provide the opportunity for anyone to use low cost solutions that are customizable based on various needs. Additional testing is always needed but there are plenty of resources and products available.

REFERENCES

AccessData. (2016). *AccessData Product Download.* Retrieved from http://accessdata.com/product-download

Bearnes, B. (2015, May 04). *External Drive as Raspberry Pi Root.* Retrieved from Adafruit: https://learn.adafruit.com/external-drive-as-raspberry-pi-root?view=all

Carlton, G. K. (2014). A Study of Forensic Imaging in the Absence of Write-Blockers. *Journal of Digital Forensics, Security and Law, 9*(3), 51-58.

Carrier, B. (2003, September). *Open Source Digital Forensic Tools - The Legal Argument.* Retrieved from Digital Evidance: http://www.digital-evidence.org/papers/opensrc_legal.pdf

Champlain College LCDI. (2016, February 16). *Raspberry Pi Forensics.* Retrieved from http://www.champlain.edu/Documents/Raspberry%20Pi%20Forensics%20(1).pdf

Convert-Me. (2016). *Convert - Data Transfer Rate.* Retrieved from http://www.convert-me.com/en/convert/data_transfer_rate/

DC3 / Defense Cyber Crime Institute (DCCI). (2016). *DC3 Tools.* Retrieved from DC3 / Defense Cyber Crime Institute (DCCI) / Tools: http://www.dc3.mil/tools

Forensicswiki - Category: Disk Imaging. (2016). Retrieved from http://forensicswiki.org/wiki/Category:Disk_Imaging

Foundation, R. P. (2016). *Raspberry Pi 2 Model B.* Retrieved from RaspberryPi.org: https://www.raspberrypi.org/products/raspberry-pi-2-model-b/

Gladyshev, L. T. (2015). Open Forensic Devices. *Journal of Digital Forensics, Security and Law, 10*(4), 97-104.

Guidance Software. (n.d.). *Encase Forensic Imager.* Retrieved from Guidance Software: https://www2.guidancesoftware.com/resources/Pages/doclib/Document-Library/EnCase-Forensic-Imager.aspx

Hardkernel. (2015). *Odroid-XU4.* Retrieved from Hardkernel: http://www.hardkernel.com/main/products/prdt_info.php?g_code=G143452239825

Hardkernel. (2016). *Hardkernel Odroid C2.* (Hardkernel) Retrieved from http://www.hardkernel.com/main/products/prdt_info.php

Hardkernel. (n.d.). *Hardkernel - Odroid XU4 Cloudshell.* Retrieved from http://www.hardkernel.com/main/products/prdt_info.php?g_code=G143599699669

Kali Linux Official Documentation - Kali Linux - Raspberry Pi. (n.d.). Retrieved from http://docs.kali.org/kali-on-arm/install-kali-linux-arm-raspberry-pi

Knight, D. (n.d.). *DietPi.* Retrieved from http://dietpi.com/

LCDI, C. (2015, 12 13). *Raspberry Pi Forensics Update - Computer & Digital Forensics Blog.* Retrieved from http://computerforensicsblog.champlain.edu/2015/12/13/1623/

Lemaker. (2016). *Lemaker Banana Pro Specifications.* (Lemaker) Retrieved from http://www.lemaker.org/product-bananapro-specification.html

Morra, S. (2013, December). *Confirming the Integrity and Utility of Open Source*

Forensic Tools. Retrieved from Utica Online: http://programs.online.utica.edu/pdf/Morra_6_Gonnella_Confirming_the_Integrity_and_Utility_of_Open_Source_Forensic_Tools_December.pdf

Morrissy, D. B. (2014). Digital Forensic Evidence in the Courtroom: Understanding Content and Quality. *Northwestern Journal of Technology and Intellectual Property*, 121-128.

MSuhanov Linux Write Blocker. (2016). Retrieved from Github: https://github.com/msuhanov/Linux-write-blocker

NetworkDLS - DiskMark. (n.d.). Retrieved from http://www.networkdls.com/Software/View/DiskMark

NIST & US Department of Homeland Security. (2013, December 27). *Test Results for Digital Data Acquisition Tool: DCFLDD 1.3.4-1.* Retrieved from https://www.dhs.gov/sites/default/files/publications/DCFLDD%201%203%204-1%20Test%20Report_updated.pdf

odroid XU4 with Kali 2, read only file system. (2015, 8 2). Retrieved from Kali Linux Forums: https://forums.kali.org/showthread.php?26431-odroid-XU4-with-Kali-2-read-only-file-system/

Offensive Security. (2016). *Kali Linux Arm Images.* Retrieved from https://www.offensive-security.com/kali-linux-arm-images/

Polstra, D. P. (2016). *PPolstra.* Retrieved from http://philpolstra.com/Home/

Polstra, P. (2014). *Hacking and Penetration Testing with Low Power Devices.* Syngress.

Raspberry Pi Forums - Any negative impact with setting `max_usb_current=1`? (2015, February 15). Retrieved from https://www.raspberrypi.org/forums/viewtopic.php?f=29&t=100244

Runge, K. (n.d.). *X11VNC.* Retrieved from http://www.karlrunge.com/x11vnc/

Sims, G. (2014, November 25). *ARM vs X86 – Key differences explained!* Retrieved from Android Authority: http://www.androidauthority.com/arm-vs-x86-key-differences-explained-568718/

Sumuri. (2016). *Sumuri Paladin.* Retrieved from http://www.sumuri.com/product-category/paladin/

TedTechnology. (2016, March 21). *SourceForge - Quick Hash GUI.* (SourceForge) Retrieved from https://sourceforge.net/projects/quickhash/

Tobin, L. (2013). *Github Firebrick.* Retrieved from https://github.com/leetobin/firebrick

US Energy Information Administration. (2016). *Electricity.* Retrieved from EIA.gov: https://www.eia.gov/forecasts/steo/report/electricity.cfm

Webopedia. (2016). *What is a Single Board Computer (SBC)? Webopedia Definition.* Retrieved from http://www.webopedia.com/TERM/S/sbc_single_board_computer.html