NISTIR 7276

The Impact of RAID on Disk Imaging

Steve Mead
Software Diagnostics & Conformance Testing Division, ITL
National Institute of Standards and Technology

NIST
National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

NISTIR 7276

The Impact of RAID on Disk Imaging

Steve Mead
Software Diagnostics & Conformance Testing Division, ITL
National Institute of Standards and Technology

July 2005

U.S. DEPARTMENT OF COMMERCE
Donald L. Evans, Secretary
TECHNOLOGY ADMINISTRATION
Phillip J. Bond, Under Secretary of Commerce for Technology
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
Arden L. Bement, Jr., Director

Title: Impact of RAID on Disk Imaging

Author: Steve Mead, Computer Scientist, NIST

Keywords:
RAID, Disk Imaging, Redundant Array of Inexpensive Disks, Forensic Examination.

Table of Contents

Table of Figures

List of Acronyms:

ATA: Advanced Technology Attachment
BIOS: Basic Input/Output System
CD: Compact Disk
CD-ROM: Compact Disk Read Only Memory
CFTT: Computer Forensic Tool Testing
CPU: Central Processing Unit
CRC: Cyclic Redundancy Check
CRC32: Cyclic Redundancy Check, length of 32 bits
DSP: Digital Signal Processors
DVD: Digital Video Disk
DVD-ROM: Digital Video Disk Read Only Memory
EIDE: Enhanced IDE
FAT: File Allocation Table
FAT16: File Allocation Table, 16-bit version
FAT32: File Allocation Table, 32-bit version
GB: Gigabyte, 1024 Megabytes
IDE: Integrated Drive Electronics
ITL: Information Technology Laboratory
JBOD: Just a Bunch of Disks
KB: Kilobyte, 1024 bytes
MB: Megabyte, 1048576 Bytes
MD5: Message Digest Version 5
NIST: National Institute of Standards and Technology
NTFS: Windows NT File System
OS: Operating System
PATA: Parallel ATA
POST: Power-On Self Test
RAID: Redundant Array of Independent Disks
RAM: Random Access Memory
SATA: Serial ATA
SCSI: Small Computer System Interface
SHA1: Secure Hash Algorithm, 160-bit message digest
UNIX: A family of Multi-user operating systems

1.0 Introduction[±]:

Forensic investigators are encountering Redundant Arrays of Inexpensive Disks (RAID) systems with increasing frequency as businesses elect to utilize systems that provide greater data reliability. RAID technology provides greater data reliability through redundancy—data can be stored on multiple hard drives across an array, thus eliminating single points of failure and decreasing the risk of data loss significantly. RAID systems often dramatically increase throughput of both reading and writing as well as overall capacity by distributing information across multiple drives.

Not only are investigators finding RAID configurations in business environments, but also in non-business settings as well. There are several reasons for this; first, RAID technology is becoming less expensive and easier to manage. Second, the nature of home computing is changing rapidly, with many users relying on computer systems for the storage of a large quantity of multi-media files (images, video, and music).

The initial and most critical aspect of a forensic examination is the complete and accurate acquisition of the target media. National Institute of Standards and Technology (NIST) was tasked with the responsibility of developing specification and testing methodologies to assure the correct functioning of disk-imaging software. Specifications and test cases were developed after identifying the common situations under which disks are imaged, which at that time did not include RAID. However, as RAID has become more common, the interaction and potential impact that RAID may have on disk imaging must be examined.

Since this research pertains to a wide audience with varied experiences, background information regarding computer forensics, disk imaging, and RAID technology is provided.

The experimental process is explained in the body of the article, with details regarding individual elements of the experiment provided in the two appendixes. Results and analysis derived from the experiments are provided which detail the relationship of RAID and disk imaging.

The paper is concluded with a summarization of findings and their impact on disk imaging, as well as recommending changes in the NIST disk imaging procedures.

1.1 Defining Computer Forensics

Forensics is the application of sciences (typically natural and physical) to the resolution of legal matters. Typically, forensics involves the identification, preservation, and analysis of material that is presented as evidence in a court of law.

[±] Certain commercial software, equipment, instruments, or materials are identified in this paper to foster understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

Forensics is not limited to the biological sciences; many fields of forensics have the same goal of identifying, preserving, and analyzing material (evidence) to be used in legal proceedings. A major, and expanding, sub-field of forensic science is that of computer forensics.

There have been many definitions proposed for computer forensics. Some focus on the general process of retrieving and chronicling evidence located on a computer hard drive. Other definitions are more specific and state computer forensics is the scientific examination, retrieval, and analysis of data contained on computer media, where computer media is defined as any device upon which a computer can store data. Across all definitions of computer forensics (and forensics in a broader scope), there are common elements which can be identified.

1. Identification - Identifying not only the physical location of the media, but also the type of media as well as the method used by the computer to store data on this media.

2. Collection -Copying content of the media onto another media source, thus allowing for further examination and analysis without endangering the integrity of the original media.

3. Analysis – Scrutiny of the evidence.

4. Presentation – Prepare the findings in an organized, methodical, and accurate manner so they can be used in a court of law as evidence.

This study examined the tools and processes used by investigators to obtain the bit-bit[1] copy of data from RAID configurations, and individual hard drives used in RAID configurations. This study also examined the imaging tools, with the specific intent of determining if such tools are impacted by a RAID.

RAID arrays are unique from other computer hard drives. Different RAID formats and configurations may impact disk imaging. For the collected evidence to be admitted in a court of law, it must be proven the data in question is an exact duplicate of the source data containing no alterations. Depending on actual configuration settings, RAID can affect both the completeness and accuracy of making an image of the target media, which potentially would provide opposing counsel an opportunity to challenge evidence due to discrepancies of imaged RAID devices.

1.2 Goals of the Computer Forensic Tool Testing Project

The Computer Forensic Tool Testing (CFTT) project[2] was designed to assess computer forensic tools used by law enforcement. Specifically, the aim was to ensure these tools produce accurate and objective results. The CFTT project established a methodology for

[1] A bit-bit match in relation to disk imaging means that for every bit of the target disk (duplicate) there is a corresponding bit in the original or source image.

[2] The CFTT project can be found at http://www.cftt.nist.gov.

testing forensic software tools through the development of specifications and requirements, testing procedures, criteria, testing sets and hardware.

The first forensic function addressed by the CFTT project was disk imaging. For tools that image hard drives (and some other forms of media), a complete set of specifications were developed to allow NIST to measure and assess the functioning of the imaging tools.

The disk imaging project first tested imaging Enhanced Integrated Drive Electronics (EIDE) and Small Computer Systems Interface (SCSI) hard drives. This research paper explores and documents the relationship between RAID and disk imaging based on controlled experiments, and makes recommendations regarding disk imaging specifications and requirements, including recommended changes to current methodologies. Additionally, the results provide investigators additional information on the potential impacts RAID technology has on imaging, and how it may affect the completeness or accuracy of the acquired images.

1.3 Technical Overview

This section provides a brief technical overview on disk imaging as well as the various aspects of RAID.

1.4 Understanding and Defining RAID

RAID technology provides a transparent way to combine multiple disks together to provide enhanced performance and/or reliability. This leverages the ability of the RAID to perform disk actions in parallel, to increase the overall throughput of data, or to increase the redundancy of information.

RAID is usually implemented in one of two ways. The most common implementation is hardware based RAID, although this method appears to be decreasing in its usage as the onboard Central Processing Unit (CPU) processing power increases. A dedicated disk drive controller is established to work with multiple disks and presents this array as a single drive to the Basic Input/Output System (BIOS) and operating system (OS). The second method used is a software based raid controller. All array functions and management functions are controlled by the software operating on the host and consume system resources such as CPU cycles and memory. The host operating system creates a virtual device that operates between the disks and the operating system drivers. This RAID device makes all participating drives appear as a one to the host operating system.

It is important to note, in a hardware-based RAID, none of the individual drives are visible, and only the hardware RAID volume is visible. In software based RAID volumes, the operating system has access to all drives as well as the constructed RAID volumes.

Depending on the amount of redundancy and performance needed, RAID has a variety of types available. For example, there are RAID mirrors (RAID-1), where the data is completely redundant on separate disks. Other common types are RAID-0, where disk striping is used to increase throughput with no redundancy; and RAID-5 which combines parallel writing with parity, providing redundancy and increased performance.

1.4.1 RAID Types Overview

Although there are multiple types of RAID, only three are primarily used; RAID-0, RAID-1, and RAID-5.

RAID-0 (Striping) is where the data is spread in a round-robin fashion between all participating disks. According to the original definitions of RAID in the Berkeley Papers[3], this form of RAID was not originally considered part of RAID, as there is no redundancy. However, because of its low overhead and parallel write strategy, it is the fastest in performance for both reading and writing, and in most hardware and software implementations it is referred to as RAID-0. As the data is written to the drive it is divided up into sequential blocks, and each block is written to the next available drive in the array, see Figure 1.

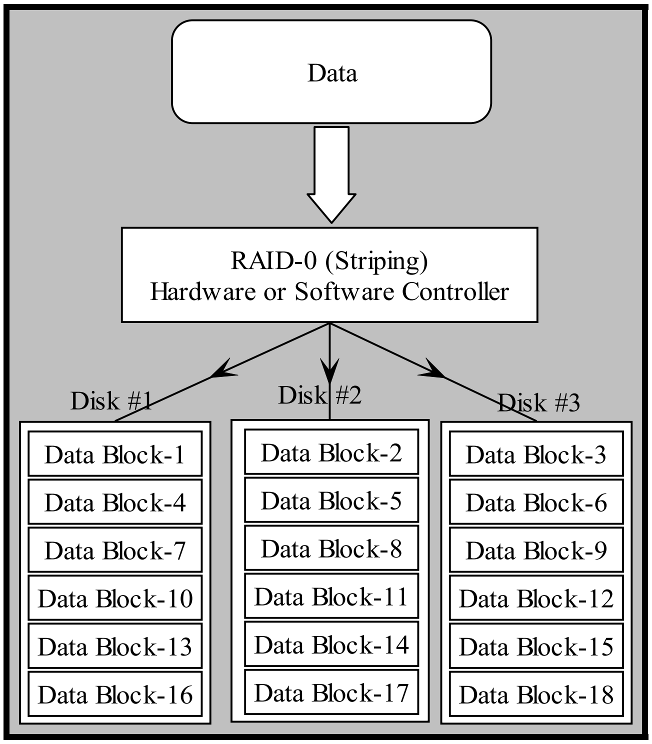


Figure 1: RAID-0 (Striping)

RAID-1 (mirroring) has the greatest amount of redundancy, as there are multiple complete copies of the data at all times. In this RAID configuration, the data is divided into sequential blocks, and each block of data is redundantly written onto each drive

[3] The Berkeley papers were a series of Papers written at the University of California, Berkeley in the late 1980's, starting with "A case for Redundant Arrays of Inexpensive Disks (RAID)", by David Patterson, Garth Gibson, and Randy Katz.

participating in the mirror. Since it is required to maintain identical copies on separate drives, RAID-1 is the slowest in terms of write performance. The RAID-1 can have fast read performance since data can be read from multiple drives in parallel, overcoming limitations in the drive channel bandwidth. Depending on the number of drives participating in the array, RAID-1 offers the highest level of redundancy, and if configured correctly can survive multiple (N-1) drive failures. For example, if there are three drives participating in a RAID-1 system, up to two drives (3-1) could fail without incurring data loss, see Figure 2.

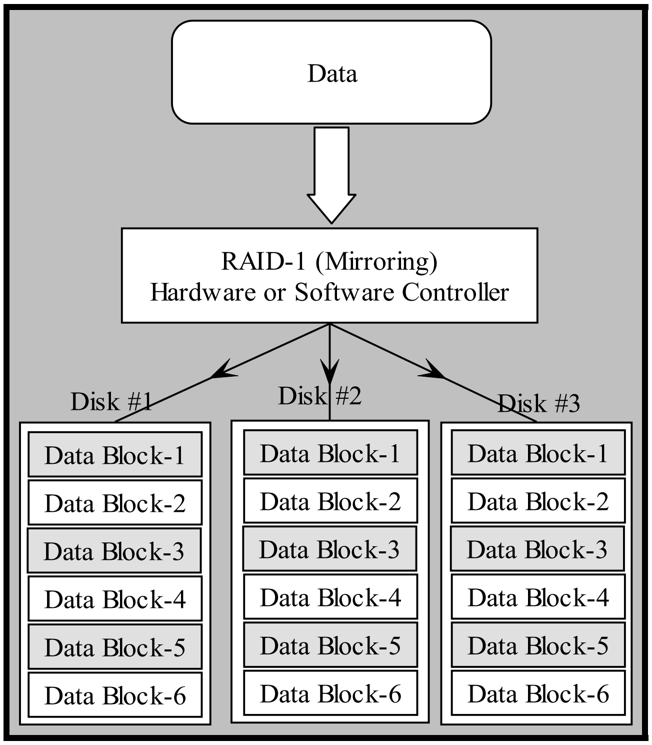
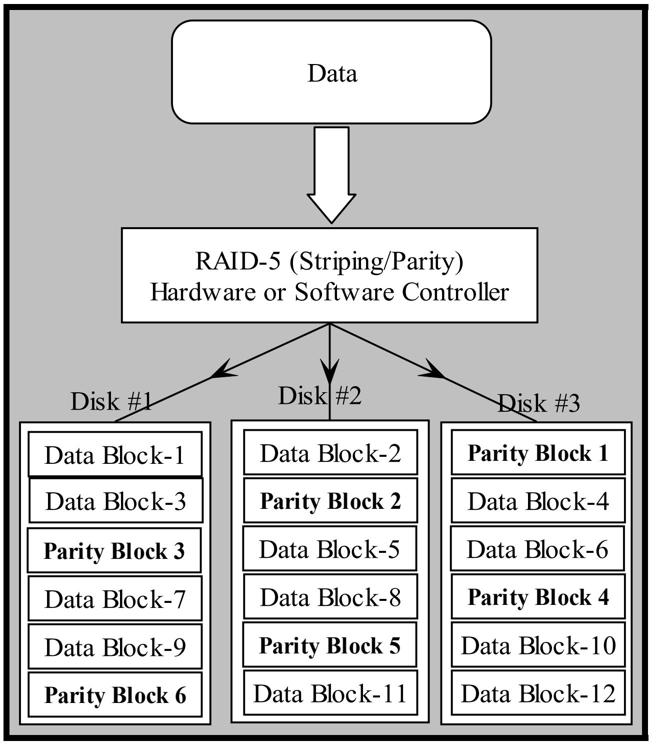


Figure 2: RAID-1 (Mirroring)

The last form of RAID commonly used is RAID-5. This combines a distributed form of redundancy with parallel disk access. Overall, this creates an array that has both high read and write performance as well as protection against drive failure. The redundancy is created by using a parity block that is distributed across all drives in the array. Since this parity information is spread across all of the drives, if a single drive should fail, any missing data can be recreated by using the remaining information and parity information. The reading and writing performance is similar to the RAID-0; the data is both read and written in parallel across all participating drives. The amount of storage space is reduced slightly due to the parity information, giving a total disk space of (N-1) drives. For example, if there are 5 drives in the RAID-5 array, then the storage space is equal to 4 drives, as the redundant parity information takes up 1/5 of the space. A RAID-5 can recover from at most one simultaneous drive failure. See Figure 3.



Disk #1	Disk #2	Disk #3
Data Block-1	Data Block-2	**Parity Block 1**
Data Block-3	**Parity Block 2**	Data Block-4
Parity Block 3	Data Block-5	Data Block-6
Data Block-7	Data Block-8	**Parity Block 4**
Data Block-9	**Parity Block 5**	Data Block-10
Parity Block 6	Data Block-11	Data Block-12

Figure 3: RAID-5 (Striping/Parity)

These three types of RAID (0, 1, 5) are the most common and almost all vendors make either hardware or software RAID solutions which include them. However, even if the actual types of RAID are the same, there are important differences between the hardware and software implementation among various vendors.

1.4.2 RAID Implemented through Hardware

The original implementation of RAID was based on proprietary hardware controllers consisting of multiple drives combined into an array transparent to the overlying operating system and file system. Initially, RAID controllers were based using SCSI drives, but currently all common forms of drives are supported Parallel-ATA (PATA)[4], Serial-ATA (SATA)[5], and SCSI. There are onboard Digital Signal Processors (DSP) to offload the work from the CPU, which allows a highly efficient means to maintain the RAID with minimal additional overhead on the host CPU.

From the perspective of the BIOS and operating system, controller cards present the RAID array as a single drive, except in the case of JBOD[6]. In the boot up process, the controller's bios checks the configuration of the RAID and allows for changes. Then, as the motherboard BIOS gathers drive information, the RAID controller presents the array

[4] Parallel-ATA (PATA) is a common interface for connecting a variety of storage devices such as hard drives, CD-Rom's, and DVD's.

[5] Serial-ATA (SATA), is the latest interface for connecting storage devices, and was designed to support much greater throughput than the original PATA interfaces.

[6] JBOD is "Just a Bunch of Disks," where the RAID controller presents drives in the array as separate disks that the BIOS can see.

as a standalone drive. Frequently there is operating system specific software that allows for on the fly monitoring and changes by the host operating system.

1.4.3 RAID Implemented through Software

When early versions of Software based RAID were released, there were significant problems with it. These RAID configurations tended to be slow due to the overhead on the CPU. At that time (the early 1980's), the processing power of CPU's were minimal and any additional overhead, such as maintaining a software based RAID, significantly impacted system performance. Additionally, the majority of implementations of software RAID volumes were contained within proprietary systems that made it difficult to examine, modify, or utilize RAID in any effective wide-scale practice.

Currently, in most cases, software based implementations of RAID no longer suffer from these limitations. The processing power of current CPU's are such that the majority of software based RAID have a negligible impact on overall computer performance. In some cases, the performance of software RAID outperforms lower end hardware implementations[7]. Additionally, there are open source operating systems such as Linux, which have non-proprietary implementations of RAID. There are still proprietary operating systems such as Windows NT/2K/XP, as well as versions of UNIX that contain their own implementations of RAID (e.g. dynamic disks in Windows environment).

For the purposes of disk imaging, the primary difference between the hardware and software RAID is that software RAID allows individual drives to be accessed through both the BIOS and the Operating system. For example, under the Linux software implementation of RAID-5, even with the array configured and working, various tools such as dd and fdisk can still see the individual drives and manipulate them. Additionally, on boot up the BIOS also recognizes each drive and provides an opportunity to adjust the drive parameters.

1.5 Computer Forensics and RAID Disk Imaging

The initial process in computer forensics is the identification and acquisition of digital data. Computer forensic investigators must document procedures taken to retrieve the data. Ideally, to provide the court confidence the data is an exact, unaltered replication of the original data in question.

When working with RAID arrays, the initial steps are still the same—to identify and obtain a complete and accurate disk image. To be complete and accurate in the eyes of the court, data must be verified as bit-bit match. Failure to provide the court assurance of data integrity can result in the evidence being completely dismissed, or used in a lesser capacity as an artifact, finding, or as an item of note.

1.5.1 Acquiring an Accurate Disk Image

[7] Initial tests were done comparing software RAID of types (0,1,5) with hardware RAID. In some cases the software raid had slightly higher throughput. It is important to note that the throughput tests were done during the configuration process of the RAID—primarily to verify the RAID controllers worked correctly and drives worked as RAID devices.

To acquire a disk image, computer forensic investigators use what is known as a disk imaging tool. This tool is designed to make a complete and accurate copy of the source media onto the destination media.

Any tool used to duplicate the data must not change or alter the source data in any way. This can be verified by computing the hash value of an entire drive (meaning the hash function is applied to all contents on the drive) both before and after the duplication procedure on the source drive. The hash value of the source drive should remain the same after duplication, which will verify the content has remained unchanged. If data has been duplicated on the same type of drive, a hash value can be obtained on that drive as well, and the resulting hash value should be identical to the hash values obtained from the source drive.

RAID has a high level of transparency, meaning the data is widely accessible to non-proprietary tools. Either hardware or software RAID should be accessible as a “normal” hard drive. Theoretically, RAID should not have any affect on disk imaging in the vast majority of cases, and should be indistinguishable from any other type of disk type or file system. Accuracy is only subject to the limitations of the imager.

In reality, due to the nature of a RAID systems operation, the apparent accuracy of disk images can be impacted. For example, if a RAID is imaged through the hardware controller the resulting image will most likely be different than if each participating hard drive is imaged separately. This is true regardless of the type of RAID used.

In the most common case, where a multi-disk array (RAID-0, RAID-5) is imaged, the data is spread across all participating drives (recall RAID-1 makes each drive an identical copy). If the drives are imaged separately, the individual drive hashes will have little resemblance to the original RAID volume. Likewise, if the RAID is imaged as a constructed and intact volume (i.e. through the RAID controller), the resulting hash will be different when compared to the hashes of each individual drive.

Even in the case of a RAID-1 (drive mirroring), where all data placed on the drives is replicated across all participating drives; there are potential problems with accuracy. If only the data partition is hashed bit-bit, each drive will hash the same—assuming both drives used in the mirror were forensically wiped before being attached to the RAID-1 mirror. However, whole disk hashes will not match, as the RAID places some unique information on each participating drive that will cause the hashes to differ.

1.5.2 Acquiring a Complete Disk Image

Accuracy is not the only aspect that may be potentially impacted when imaging a RAID configuration. Completeness is the second fundamental aspect of disk imaging, and verifies the tool duplicates all data on the source media. RAID, in theory, should provide a transparent view of the source data without impacting current forensic tools. This is normally validated by verifying every bit of data on the source has a corresponding bit on the duplicated drive. Additionally, using a hash function also provides a means to ensure that all data has been duplicated.

An imaging tool should create a complete image of a given target media. In the RAID environment, this causes a slight problem, as there are situations where a disk imaging tool works correctly when imaging the RAID volume yet does not copy all of the information on a specific drive within the constructed array. There are two cases imaging tools encounter; one affects the completeness of the image, and the second does not. The first is a case that requires a volume to be imaged through a RAID hardware controller. The second case is where each discrete hard drive to be imaged separately (i.e. without going through a RAID controller), and can be a software based RAID, or by physically removing the drive for imaging.

In the first case, the RAID is based on a hardware controller, and a forensic-imaging tool is used on the visible volume. The imaging tool can only see the active RAID volume as a whole, and does not have any access to any individual drives participating in the array. From the perspective of the imaging tool, the RAID volume accessed through the hardware controller can be imaged completely. The partition can be copied bit-bit, and a verification hash can be computed for all the data imaged. However, this image is only what the hardware controller allows to be accessed, and none of the drives participating in the RAID are imaged in their entirety.

For example, let's examine this hypothetical situation. Suppose you have two 20 GB drives configured through a hardware RAID-1 mirror which are configured through the RAID controller BIOS. The size of the RAID-1 mirror could be less than the maximum values of the drives—so in this case, the RAID-1 partition could be 19 GBs, 10 GBs, or 1 GB. The imaging tool would only be able to see the active volume, and would not have access to any other parts of the disks—creating a complete RAID volume image, but not complete drive images. This would be a common example if mismatched drives were used to construct the array, as the smaller drive would limit the maximum size of the RAID-1 volume.

Essentially if the image was constructed with all of the hardware and drives in tact, the imaging tool may not capture all data across all drives participating in the RAID array, but it will image the accessible RAID volume completely.

Regarding the second case, where entire RAID drives are imaged separately (as per standard forensic practice), it does not matter if the RAID is hardware or software based as the images will be complete. The individual images are subject only to the limitations of the imaging tools used. Essentially, imaging the RAID drives separately is no different than imaging any other type of drive format such as PATA, SATA, or SCSI.

1.6 Digital Data Verification (Hashing)

In the field of computer forensics, during the disk imaging process, two of the most critical factors are obtaining a complete disk image, and obtaining an accurate disk image. One of the main methods to ensure either, or both of these properties is through using a hashing algorithm.

A hash is a numerical code generated from a stream of data—considerably smaller than the actual data itself, and is referred to as a message digest. It is created by processing all of the data through a hashing algorithm, which generates a fixed length output. Due to the properties of a good hashing algorithm, it is computationally infeasible to have any other different stream of data produce the same hash output. When a duplicate is made of a disk, the bits can be passed through the hashing algorithm to create a unique value, which can be compared against the original and any other subsequent copies.

As mentioned above, the hash is based on a mathematical formula and is created by what is known as a hash function. Currently, there are three major algorithms used in computer forensics as hash functions:

- CRC-32 – This is actually not a hash function, and in reality is too weak to be heavily relied upon. CRC-32 is actually a 32-bit checksum. The main weakness is that the probability of two separate and distinct data-streams generating the same value using CRC-32 is too high.

- MD5 – MD5 is a 128-bit hash algorithm, and is not susceptible to the same weakness of CRC-32. The chances of any two distinct data-streams generating the same hash value using MD5 is extremely low.

- SHA1 – This is a 160-bit hash algorithm, which is computationally stronger than the MD5.

Although hash functions can be used to verify that a disk image has been created completely and accurately, they can also be used to verify any other bits of data, such as files and program output. Specifically in relation to disk imaging, the benefit of using a hash algorithm is that if any bit is changed or missing between the source and the destination copy, a hash of the data-stream will show this difference.

1.6.1 Differences between Imaging RAID and Independent Drives

Regardless of whether a hardware or software RAID configuration is used, the result is the same—both methods increase redundancy as well as performance. However, from the perspective of disk imaging tools, and hence from the point of view of the computer forensic imaging tool, there are a three key points to keep in mind.

First, hardware RAID systems do not allow direct access to any of the drives participating in the array—either from the BIOS or through the operating system. This directly affects the disk-imaging tool. A constructed RAID volume and the controller do not allow independent access to any of the participating drives. Even when the host operating system is bypassed by using boot disks (e.g. DI-Tool-#2 floppy boot), the drives present on the system are only accessible through the RAID controller. To help put this in perspective, while a forensic investigator may physically observe a 15 disk RAID-5 array, from the perspective of any disk imaging tool on that system there is only one drive present.

Second, if an operating RAID volume is acquired by imaging each disk participating in the array, the hashes will be different. For RAID arrays that write in parallel (RAID-0, RAID-5), this is to be expected as the data is not the same on each drive. However, this is true even in the case of mirrored drives (RAID-1), since there is additional information written outside of the visible partition that will change the overall drive hash—although the data partition between the drives may hash the same.

Third, there must be a clear understanding of how the RAID will be reconstructed after the imaging process. If the intact RAID volume is imaged onto a non-RAID host media, then no additional processing needs to take place. However, if a RAID is imaged through individual drives, additional steps must be taken to be able to correctly image, verify and reconstruct the RAID. For hardware based RAID, a complete inventory should be taken of all RAID hardware, drive configurations, and the RAID controller setup, so the RAID can be physically reconstructed. Even in the case of software RAID, a complete inventory should be taken since the implementation of RAID may be proprietary, and depend on a given operating system.

2.0 Expected Findings

Initially it was not suspected RAID would have an impact on the accuracy or completeness of disk imaging. This was expected to hold true for disk images obtained through both software and hardware RAID configurations. It was recognized in some instances RAID hardware controllers could affect the completeness of disk imaging since they do require a minimal amount of physical storage space on hard drives for RAID management, and experiments were developed to focus on these special cases. However, any impact on the actual data obtained from disk imaging was considered negligible.

2.1 Overview

The testing procedure and test cases were derived from standard testing methodology. All factors of interest were used to develop a matrix, from which the various cases were enumerated. However, during the initial exploratory testing, the testing results suggested that under most conditions, RAID did not affect disk-imaging tools. This was supported through additional research; since RAID is essentially a structure lying below the file system, it does not fundamentally change the acquisition of target media.

At this point of the research, the overall testing methodology shifted to focus on finding boundary conditions where RAID did indeed affect disk-imaging tools. During this phase of testing, some additional, interesting, interactions were observed which were subsequently examined in detail. It is important to note the primary focus of this research focused on disk imaging aspect of RAID systems, and not on reconstructing a RAID after acquisition.

2.2 Initial Testing Assumptions

Although there are a variety of RAID types, combined with the hardware or software implementation, disk imaging tools are only affected in a limited way. According to research and some initial experimental analysis, the following pre-established assumptions were used.

- The RAID type (0, 1, and 5) has no impact on the actual process of drive imaging—if individual drives are imaged.

- Software constructed RAID has no impact on the drive imaging. The imaging tools are designed to image individual drives. Under a software RAID, the drives are accessible to BIOS, the OS, and consequently the imaging tool.

- Hardware RAID is the only case where the imaging tool is impacted. In this case, the RAID is presented through the controller as a single drive, eliminating the BIOS and OS from having access to the individual drive. The imaging tool will be able to do a bit-bit RAID volume image, but depending on how the RAID is constructed, it may be incomplete, as it will not image all bits on the target drive.

- If all drives participating in a RAID are imaged separately, then it does not matter if the RAID was hardware or software based, as the imaging tool has access to all the data on the drive.

Ideally, by maintaining and adhering to standard forensic practices where each drive is imaged separately, imaging tools are not affected. However, after the imaging process, it is important to note that reconstructing the RAID could be problematic without additional information on how the source RAID was originally implemented.

Initially a complete testing matrix was created with all relevant factors used to generate all of the testing cases (conformance testing perspective). However, this methodology changed as more research was conducted. RAID is designed to work within/under file systems, and should be transparent in most settings. As a result, the experiment was divided into two main parts. The first part was a verification that disk imaging works correctly on a RAID volume. The second set of experiments was designed to focus on identifying boundary conditions and areas where RAID affects disk-imaging tools. There were some additional changes made as well to simplify the experiments, to reduce the overlapping of the RAID types and its hardware or software implementation. The use of SCSI or PATA devices in RAID configurations was also examined.

2.2.1 Tested RAID Types

Even though there are multiple RAID types, only three, 0, 1, and 5 are commonly used and implemented in the majority of RAID hardware and software applications. Moreover, even those types can be broken down into two general categories; mirrored (‘single-disk replication’), and multi-disk RAID. RAID-1 (mirrored) is where the drives participating in an array or direct bit-bit copies of each other, and a single participating drive can be imaged without needing to image the whole RAID. The other two types of RAID (0 and 5), are multi-disk images, where all the participating drives need to be imaged to be able to reconstruct the data for analysis.

For this study, the RAID types are collapsed into two cases, disk mirroring, or multi-disk—specifically for testing purposes will be either a mirrored array (RAID-1), or multi-disk array (RAID-5).

- "Single" drive RAID volumes, where data is duplicated across all participating drives, as would be the case with RAID-1 mirror.

- Multi-Disk RAID volumes where the data is written in parallel across multiple drives with parity (RAID-5) or without (RAID-0).

The experiments are conducted using two RAID types with both SCSI and ATA drives. This enables us to examine and compare the results, including specific details regarding disk mirroring and multi-disk RAID configurations.

2.2.2 RAID Hardware/Software Configuration

The creation and maintenance of a RAID affects how the drives should be imaged. If a hardware RAID is imaged in place with the drives and controller installed, there may be data on the individual drives that is not captured through the imaging process. However, if the drives are pulled out and imaged separately following standard forensic procedures, then RAID should not influence the imaging process.

For testing, we used three different RAID controllers, the first two are by Adaptec, and the last one is by Promise. For software RAID configurations, the drives have no special configuration, as the RAID is constructed through software.

2.2.3 Drive Types (PATA, SCSI)

Originally, there were distinctions between SCSI and PATA interfaces for RAID. After some initial testing, the type of device did not affect the imaging process. Additionally, the imaging tools were previously tested under PATA and SCSI devices, and were not affected by the interface type. There are no specific case examples to distinguish between the interfaces, but both PATA and SCSI will be identified and used within the testing process.

2.2.4 Imaging Tools

The disk imaging tools used to test the impact of RAID were those which have been previously used in the CFTT disk imaging project. Primarily this was to leverage our experience in the configuration, and usage of disk imaging tools. Additionally, this also provided us with a good set of previous tests and outcomes that we could use to compare against the current set of RAID imaging tests.

It is important to point out that since we are concerned with the interaction between RAID and disk imaging and not on the performance of imaging tools, we have removed references to specific tools and refer to them as DI-Tool #1 through DI-Tool #3.

Primarily this was to leverage our previous experience with the tools in the imaging. Additionally it provided a good comparison for the RAID disk imaging tests.

DI-Tool-#1:
This imaging tool uses a boot disk to image the target device. The RAID host system was booted using the disk, and imaging took place under the boot disks' direction.

DI-Tool-#2:
This tool has a disk imaging function, and is initiated using a Bootable CD-Rom downloaded from the developer's website. The configuration for this tool was different than the first, as we transferred the image across a dedicated network cable to a secondary machine for acquisition and analysis.

DI-Tool-#3:
The third imaging tool comes as a standard utility in a variety of operating systems, and used by forensic examiners to image hard drives. The RAID host is booted using a bootable hard drive, and imaged directly onto a dedicated storage drive.

2.2.5 Additional Tools

There were two additional tools used in our testing. The first is Partition Magic, and the second was DiskHash (a CFTT Tool). Partition Magic was used mainly to setup partitions, scan them for bad sectors, and then wipe them with a known value. Additionally it provided additional information on the partitions, like size, and partition type. DiskHash was used to get information on hard drives and to calculate disk hashes. In certain cases for examining the actual data on the drive, or dd image, we used either HexEdit, or Norton Disk Edit.

2.2.6 Test Hardware Harness (Base Configuration)

The testing hardware consisted of the target machine, Aqua, which hosted the RAID, and in some cases an additional computer (Sepia) was be used for the analysis of the disk imaging results. To avoid conflicts with multiple RAID controllers being installed at the same time, only a single given RAID controller could be installed and active at one time.

RAID Host Hardware (Aqua):
- Intel Desktop Motherboard, D875-PB2
- Intel Pentium 4, 2.6 GBs, 1024 MB RAM
- Generic Sound Blaster Audio card
- Generic Intel Pro 10/100 Network Card
- Generic 52x CD-Rom (PATA-2: Slave)
- Boot Hard Drive—20 GB Maxtor 52049H4 (PATA-1: Master)

RAID
- Internal: Pullout Bays for removable PATA/IDE Drives
- External: Four Pull-Out Drive Bays for Removable SCSI Drives.
- SCSI #1: Adaptec SCSI RAID 2100S (0, 1, 5)
- ATA #1: Promise ATA RAID SX4000 (0, 1, 5)
- SCSI #2: Adaptec SCSI RAID 2110S (0, 1, 5)

Figure 4: General Test Harness Configuration

3.0 Execution of the Study

In the first phase of our study, expected findings were tested through standard RAID configurations using both software and hardware cases. The objective was to verify the ability to obtain a complete and accurate disk image. Any elements of RAID that were found to impact the ability of computer forensic investigators to obtain a complete and accurate image would need to be broken into boundary cases and identified to investigators.

Details of each of the cases are presented in Appendix B and C. Two different RAID controllers were tested in both RAID-1 and RAID-5 configurations. Each controller was tested for DI-Tool-#2, DI-Tool-#1, and DI-Tool-#3. In all cases, the software was able to create complete and accurate disk images based on individual drive hashes. These software packages do perform accurate images in standard cases.

The second phase of the study explored boundary cases with the goal of identifying areas where investigators may have difficulty obtaining complete or accurate images. This phase of the study also explored any other aspects of RAID imaging which were deemed to be potentially of interest to investigators.

3.1 Disk Imaging Experimental Testing (Standard Usage)

Each of the standard test cases were configured using the base system and the following general procedure was followed.

First, the drives were wiped and then installed within the RAID controller. As a quick side-note, the wiping was done using Partition Magic, and essentially consisted of writing a consistent value to every byte on the hard drive. In most tests, we used "FF" or the binary equivalent of "11111111." Each controller was then configured with the correct type of RAID (either RAID-1 or RAID-5), and was allowed to build the array if this was an option.

Second, three partitions were created on each RAID volume consisting of a 100 MB FAT partition, 100 MB FAT32, and a 100 MB NTFS partition.

Third, depending on the type of RAID, the RAID volume, and individual partitions, each drive was imaged and hash-verified before and after the imaging process. For RAID-1, we verified that the RAID-Volume was correct, and that between each mirrored drive they hashed the same. For the RAID-5 arrays we verified that the volume could be reconstructed and the original and rebuild volume matched. The general summary of each test case, and the results are in Figure 5, with complete details of each test case are in Appendix B.

Experiment ID	Raid	Controller	Disk Type	Software	Partitions Accurate & Complete	Restore Accurate & Complete
RAID-DI-TC-01	RAID 1	Promise SX4000	2 ATA Drives	DI-Tool-#2	Yes	Yes
RAID-DI-TC-02	RAID 5	Promise SX4000	4 ATA Drives	DI-Tool-#2	Yes	Yes
RAID-DI-TC-03	RAID 1	Adaptec 2110S	2 SCSI	DI-Tool-#2	Yes	Yes

RAID-DI-TC-04	RAID 5	Adaptec 2110S	4 SCSI	DI-Tool-#2	Yes	Yes
RAID-DI-TC-05	RAID 1	Promise SX4000	2 ATA Drives	DI-Tool-#1	Yes	Yes
RAID-DI-TC-06	RAID 5	Promise SX4000	4 ATA Drives	DI-Tool-#1	Yes	Yes
RAID-DI-TC-07	RAID 1	Adaptec 2110S	2 SCSI	DI-Tool-#1	Yes	Yes
RAID-DI-TC-08	RAID 5	Adaptec 2110S	4 SCSI	DI-Tool-#1	Yes	Yes
RAID-DI-TC-09	RAID 1	Promise SX4000	2 ATA Drives	DI-Tool-#3	Yes	Yes
RAID-DI-TC-10	RAID 5	Promise SX4000	4 ATA Drives	DI-Tool-#3	Yes	Yes
RAID-DI-TC-11	RAID 1	Adaptec 2110S	2 SCSI	DI-Tool-#3	Yes	Yes
RAID-DI-TC-12	RAID 5	Adaptec 2110S	4 SCSI	DI-Tool-#3	Yes	Yes

Figure 5: RAID/DI Standard Usage Test Cases Summary

3.1.1 Test Cases Summaries for Standard Usage RAID/DI

RAID-DI-TC-01: This was a PATA 2-disk RAID-1 constructed using the Promise SX4000 hardware controller, and imaged using DI-Tool-#2.

- The RAID-1 volume hashed the same before and after imaging the drives, as well as the individual FAT, FAT32, and NTFS partitions.
- Each individual (physical) drive hashed differently from each other, as well as from the RAID-1 volume mirrored within each drive.
- All reconstructed partitions and the RAID-1 volume hashed the same as the originals.

RAID-DI-TC-02: This was a PATA 4-drive RAID-5 constructed using the Promise SX4000 hardware controller, and imaged using DI-Tool-#2.

- The RAID-5 (Stripe/Parity) volume was acquired and imaged without difficulties, and restored volume hash matched the original.
- Each partition on the RAID-5 volume (FAT, FAT32, and NTFS) was imaged and the hashes matched.
- Each drive was imaged independently, and all hashes matched before and after imaging.

RAID-DI-TC-03: This was a SCSI 2-drive RAID-1 constructed using the Adaptec 2110S hardware controller, and imaged using DI-Tool-#2.

- The RAID-1 volume hashed the same before and after imaging the drives, as well as the individual FAT, FAT32, and NTFS partitions.
- Each individual (physical) drive hashed differently from each other, as well as from the RAID-1 volume mirrored within each drive.
- All reconstructed partitions and the RAID-1 volume hashed the same as the originals.

RAID-DI-TC-04: This was a SCSI 4-drive RAID-5 constructed using the Adaptec 2110S hardware controller, and imaged using DI-Tool-#2.

- The RAID-5 (Stripe/Parity) volume was acquired and imaged without difficulties, and restored volume hash matched the original.

- Each partition on the RAID-5 volume (FAT, FAT32, and NTFS) was imaged and the hashes matched.
- Each drive was imaged independently, and all hashes matched before and after imaging.

RAID-DI-TC-05: This was a PATA 2-drive RAID-1 constructed using the Promise SX4000 hardware controller, and imaged using DI-Tool-#1.

- The RAID-1 volume hashed the same before and after imaging the drives, as well as the individual FAT, FAT32, and NTFS partitions.
- Each individual (physical) drive hashed differently from each other, as well as from the RAID-1 volume mirrored within each drive.
- All reconstructed partitions and the RAID-1 volume hashed the same as the originals.

RAID-DI-TC-06: This was a PATA 4-drive RAID-5 constructed using the Promise SX4000 hardware controller, and imaged using DI-Tool-#1.

- The RAID-5 (Stripe/Parity) volume was acquired and imaged without difficulties, and restored volume hash matched the original.
- Each partition on the RAID-5 volume (FAT, FAT32, and NTFS) was imaged and the hashes matched.
- Each drive was imaged independently, and all hashes matched before and after imaging.

RAID-DI-TC-07: This was a SCSI 2-drive RAID-1 constructed using the Adaptec 2110S hardware controller, and imaged using DI-Tool-#1.

- The RAID-1 volume hashed the same before and after imaging the drives, as well as the individual FAT, FAT32, and NTFS partitions.
- Each individual (physical) drive hashed differently from each other, as well as from the RAID-1 volume mirrored within each drive.
- All reconstructed partitions and the RAID-1 volume hashed the same as the originals.

RAID-DI-TC-08: This was a SCSI 4-drive RAID-5 constructed using the Adaptec 2110S hardware controller, and imaged using DI-Tool-#1.

- The RAID-5 (Stripe/Parity) volume was acquired and imaged without difficulties, and restored volume hash matched the original.
- Each partition on the RAID-5 volume (FAT, FAT32, and NTFS) was imaged and the hashes matched.
- Each drive was imaged independently, and all hashes matched before and after imaging.

RAID-DI-TC-09: This was a PATA 2-disk RAID-1 constructed using the Promise SX4000 hardware controller, and imaged using DI-Tool-#3.

- The RAID-1 volume hashed the same before and after imaging the drives, as well as the individual FAT, FAT32, and NTFS partitions.

- Each individual (physical) drive hashed differently from each other, as well as from the RAID-1 volume mirrored within each drive.
- All reconstructed partitions and the RAID-1 volume hashed the same as the originals.

RAID-DI-TC-10: This was a PATA 4-drive RAID-5 constructed using the Promise SX4000 hardware controller, and imaged using DI-Tool-#3.

- The RAID-5 (Stripe/Parity) volume was acquired and imaged without difficulties, and restored volume hash matched the original.
- Each partition on the RAID-5 volume (FAT, FAT32, and NTFS) was imaged and the hashes matched.
- Each drive was imaged independently, and all hashes matched before and after imaging.

RAID-DI-TC-11: This was a SCSI 2-drive RAID-1 constructed using the Adaptec 2110S hardware controller, and imaged using DI-Tool-#3.

- The RAID-1 volume hashed the same before and after imaging the drives, as well as the individual FAT, FAT32, and NTFS partitions.
- Each individual (physical) drive hashed differently from each other, as well as from the RAID-1 volume mirrored within each drive.
- All reconstructed partitions and the RAID-1 volume hashed the same as the originals.

RAID-DI-TC-12: This was a SCSI 4-drive RAID-5 constructed using the Adaptec 2110S hardware controller, and imaged using DI-Tool-#3.

- The RAID-5 (Stripe/Parity) volume was acquired and imaged without difficulties, and restored volume hash matched the original.
- Each partition on the RAID-5 volume (FAT, FAT32, and NTFS) was imaged and the hashes matched.
- Each drive was imaged independently, and all hashes matched before and after imaging.

3.1.2 Findings

Our findings for the standard cases were consistent with our initial assumptions. The various RAID types had no impact on the accuracy or completeness of imaging the RAID volume. In every case, the individual partitions within the RAID volume were complete and accurate, as was the overall RAID volume.

We did find differences in the hashes between the individual drives used for a hardware RAID as compared to the actual RAID volume itself, even on RAID-1 (mirrors). This appeared to be the result of information being written to each participating drive within the RAID by the controller outside of the accessible data area. Further research provided feedback from RAID vendors indicating RAID hardware controllers tag the drives as part of the ordering process. Additionally, there is also overhead (metadata) information written any drive depending on its role within a given RAID.

Even though RAID seems to have no impact on imaging the active data portion of the drive, the differences discovered warranted further investigation. In the following section, we examine the cases where there are differences in hashes between the physical drives and the RAID volume, as well as additional research in how RAID functions, and its potential impact on the forensic process.

3.2 Disk Imaging Experimental Testing (Special Cases)

Due to some of the differences we found previously, we developed additional test cases to clarify how RAID may impact imaging. Primarily this focuses around the interaction of a hardware RAID controller, and the resulting changes made to the drives if they are examined individually (not within a constructed RAID array). The procedures varied depending on the interaction measured, and each test case is covered separately. See Figure 6.

Experiment ID	Raid	Controller	Disk Type	Problem
RAID-DI-SC-01	RAID-1	Promise SX4000	2 ATA Drives	Size differences between RAID-1 Volume and individual ATA drives.
RAID-DI-SC-02	RAID-1	Promise SX4000	2 ATA Drives	Hash differences between RAID-1 Volume and individual ATA Drives.
RAID-DI-SC-03	RAID-1	Promise SX4000	2 ATA Drives	Drive bias in residual data on Mirrored Drives during the Imaging process.
RAID-DI-SC-04	RAID-5	Promise SX4000	4 ATA Drives	Hardware RAID-5 Data and Parity Distribution.
RAID-DI-SC-05	RAID-1	Promise SX4000	2 ATA Drives	Hiding data partitions within RAID-1
RAID-DI-SC-06	RAID-5	Promise SX4000	4 ATA Drives	Examining Non-Reconstructed RAID-5 Array, individual drive data.
RAID-DI-SC-07	RAID-5	Adaptec 2100S	4 SCSI Drives	Differences in Total Sectors on SCSI RAID-5 between Partition Magic, DI-Tool-#1, and DI-Tool-#2
RAID-DI-SC-08	RAID-5	Promise SX4000	4 ATA Drives	RAID-5 Information written during 4-drive volume construction.
RAID-DI-SC-09	RAID-1	Promise SX4000	2 ATA Drives	RAID-1 information written during a 2-drive volume construction
RAID-DI-SC-10	RAID-5	Promise SX4000	3 ATA Drives	RAID-5 Information written during a 4-drive volume construction.
RAID-DI-SC-11	RAID-5	Promise SX4000	4 ATA Drives	RAID-5 Information written during a 4-drive volume construction, 16k Block size

Figure 6: Boundary Case Overview and Purpose

3.2.1 Test Case Summaries for Special Cases RAID/DI

RAID-DI-SC-01: There appeared to be a difference between the size of the constructed RAID-1 mirror image, and the actual size on each drive in the array. This held true even when each drive was identical and the RAID-1 array was maximized.

Standard setup was used, and a RAID-1 was setup using 2 ATA-drives and the Promise SX4000 controller. Configuration of the RAID was set to the maximum size (40 GBs) for the mirror, and the controller was allowed to construct the array. The size of the array, as well as the size of each individual dive was compared, and differences were found.

- We repeated the test using SCSI, ATA, and a variety of drives, and in every case, the size of the RAID volume was slightly less than the size of the participating drives.

- If one of the participating drives was smaller than the other drive, for a RAID-1, the hardware would set the "maximum" size to be the smallest of the participating drives.

RAID-DI-SC-02: In earlier experiments, there were differences in the hash values between RAID-1 mirrors, and hashes of the drives participating in the array. This experiment focused on these differences in hash values. Further examination was conducted to determine if this occurrence was due to residual information remaining on the individual drives, or was a result of unique RAID information written to the drives during the array construction process or only due to a size differential between RAID volume and the actual size of the drives.

The standard system setup was used, with a RAID-1 array constructed using 2 ATA-drives and the Promise SX4000 controller. Each drive was wiped before hand with the hex value "FF", and then placed within the RAID array and the maximum RAID-1 mirror (40 GBs) was constructed. After modifying the settings, changing write patterns, and examining each drive using hex editors, the following results were observed.

- There is a difference in the size between the RAID-1 volume and each individual drive. Since hashing is a bit-bit process, size differences, even with null values will change the resulting hash.

- Outside of the RAID-1 volume there is data that the RAID controller writes each participating drive. This data would not be hashed within the RAID-1 volume, but would be part of the hash of the individual drive, and causes the hashes to be different between the RAID-1 volume and each individual drive. Additionally, this information appears to be unique to each drive, which would also cause the hashes between mirrored drives to differ.

- Forensic hashing of a disk is bit-bit, if the original drives were not wiped before use the pre-existing residual information contained within each drive would cause the hashing to differ. If the drives are not wiped prior to use, and the RAID controller does not initialize the RAID volume, then the residual information within the RAID-1 volume could be different, causing the hashes between the RAID-1 volumes on each participating drives to differ.

RAID-DI-SC-03: In the RAID-1 documentation, reading from a mirrored RAID has the advantage of being able to read data from both drives, thus bypassing single drive throughput limitations. However, within the realm of a forensic examination, this could impact what type of residual data recovered from a drive. When a bit-bit image is performed on a RAID drive, depending on how the reads are performed, the returned residual information may be from any of the drives participating in the RAID-1 mirror.

The setup used was the standard system with a RAID-1 array constructed using 2 ATA 40GB drives and the Promise SX4000 controller, setting the maximum size of 40 GBs available for the mirror. The first drive was wiped with the hex value of "11," and the second with "FF." Out of the entire 40 GBs available, a 1 GB Fat-32 partition and a 1 GB NTFS partition were created, leaving the remaining 38 GBs as unallocated space. The controller was allowed to construct the full array. Each drive was imaged using DI-Tool-#2, DI-Tool-#1, DI-Tool-#3, and was examined with a Hex Editor. Additionally, the drives were exchanged (drive-1 moved to drive-2 slot, and drive-2 in drive-1 slot), and then the same imaging and examination was performed.

- The 1-GB Fat-32 and the NTFS partition hashed to the correct and identical values.

- Initially, the unallocated space on the RAID-1 volume for all sectors was the hex "FF," and the drive was located physically on channel 2.

- After swapping drive #1 and #2, all of the unallocated space had the hex value of "11," the drive was located on channel 2.

RAID-DI-SC-04: RAID-5 distributes data across multiple drives, as well as reduces the chance of data loss through writing a parity block as well—which can be used to reconstruct missing or incomplete data provided no more than a single drive has been damaged. This experiment was to better understand what data is transferred from each drive during RAID-5 reads, and if there is any bias on any of the drives in the constructed RAID-5.

Setup consisted of the RAID-5 being constructed using 4 ATA 40 GB drives on the Promise SX4000 controller. This gives a 120 GB RAID-5 volume, as ¼ of the space is used for parity. Each drive was wiped with a hex pattern; "11" for the first drive, "22" for the second, "33" for the third, and "44" for the fourth. After the RAID-5 configured, the Promise controller was allowed to construct the whole array. 3 small partitions were created on the drive; 100 MB FAT, 100 MB FAT-32, and a 100 MB NTFS. The drives were imaged using DI-Tool-#2, DI-Tool-#1, DI-Tool-#3, and then examined using Norton DiskEdit (Hex Editor).

- All partitions hashed correctly, and the restored partitions hashed correctly.

- Examining the unallocated space using a Hex Editor showed the following pattern as we read sectors linearly across the RAID-5 volume. In the actual results, the (P) was where there was a skip in the 11/22/33/44 pattern.
    - (p) 22 33 44
    - 11 22 33 (p)
    - 11 22 (p) 44
    - 11 (p) 33 44
    - (p) 22 33 44
    - 11 22 33 (p)

- The (P) is consistent with the parity bits being written ¼ of the time, and rotated across all drives.

RAID-DI-SC-05: Using information gained from examining the RAID-1 drive biasing, it is possible to construct an array where data could be hidden from a forensic examiner under certain circumstances.

Setting up the computer was a bit unique in this case. A RAID-1 mirror was setup with 2 40-GB ATA drives, using the Promise SX4000 controller, as well as a boot drive located on a standard EIDE controller (channel 0). Additionally, there was a spare hard drive bay available that was on standard EIDE controller—allowing for a drive from the RAID-1 array to be easily removed from the RAID array and plugged into a standard (non-RAID) EIDE channel.

Initially, the RAID-1 was setup and configured using a 20 GB FAT-32 volume on each of the ATA drives. Then the machine was shut down, and one of the drives was moved to the standard EIDE channel. The RAID-1 was disabled, the machine booted, and a 5 GB partition was created at the end of the drive, files were written to it, and the system was shut down.

The drive was moved back into the RAID-1 array, and the system was booted with the RAID enabled. The RAID-1 was imaged using DI-Tool-#2, DI-Tool-#1, and DI-Tool-#3, as well as examined using Norton DiskEdit.

- If the RAID-1 bias is known (the channel that residual information is pulled from), then the non-RAID partition can be “hidden” on that drive without being detected. The RAID controller will mount only the RAID-1 volume, and ignore the other partition. When the RAID volume is imaged, any residual information will be pulled from the other drive, leaving the non-RAID partition undetected

- It is important to note, if the drives are acquired using the standard forensic practice of imaging each drive separately, all information on both drives will be acquired.

RAID-DI-SC-06: It may be the case that investigators may not have access to all of the drives in a given RAID-5, or enough of the original drives to reconstruct the entire array using the parity information. This experiment examined the drives separately to determine the possibility to conduct a forensic examination on the available drives, and what data remains on the drives in a form that is useful.

Setup of the system consisted of using a Promise SX4000 controller to configure a 4-drive, ATA RAID-5 system. After the controller was setup, 2 small partitions were created: a 100 MB Fat-32, and a 100 MB NTFS partition, as well as a variety of files and programs copied to both of the partitions (exe, txt, doc, etc…). DI-Tool-#2, and Norton DiskEdit was used to examine the disk and see what information was recoverable from the individual drives.

- There was no metadata information about the partitions, files, or any other aspect of the file system that was recoverable.

- Some readable information from text files was visible on a few of the drives, but it was difficult to identify what specific file it originated from. Primarily this was because the RAID writes information in 64k blocks.

Overall, it was difficult to do forensic examination of a non-reconstructed RAID-5 with multiple drives missing. However, there is some available information, and it might be possible to reconstruct the remaining information in a more coherent form.

RAID-DI-SC-07: During one of the experiments, there was as discrepancy in the number of total sectors being imaged from the SCSI RAID-5. We examined the issue further in this boundary case by constructing a RAID-5, and then observing differences in the size of the volume as reported by DI-Tool-#2, DI-Tool-#1, and Partition Magic.

We setup our system using the Adaptec 2100S controller, and used 4 SCSI 18.4 GB drives to create a RAID-5, and created 3 partitions (FAT, FAT32, and NTFS).

- DI-Tool-#2 and DI-Tool-#1 both reported a total of 107526144 sectors of accessible data.

- Partition Magic 6 reported 107522982 available sectors.

- DiskHash, a tool developed by NIST for the CFTT Disk Imaging project, reported 107526144 sectors.

RAID-DI-SC-08: During the process of constructing a RAID volume, not all of the available disk space is allocated or used. This is true even if the RAID is configured to use all available drive space for a given volume. This experiment is to examine the amount of space reserved, where it is located, and what data it contains.

The system is configured using the Promise SX4000 controller, and 4 ATA drives are used to construct a RAID-5. Each drive was wiped with a specific value, drive #1—"A1," drive #2—"B2," drive #3—"C3," and drive #4—"D4." The controller was used to construct a RAID-5, yielding a 120 GB RAID-5 volume. Each drive was examined using DI-Tool-#2, as well as Norton DiskEdit.

- Each drive had the first 65049 bytes initialized by the controller into 4 distinct parts.
    - The first 30464 bytes were initialized to "00" by the RAID controller.
    - The next 16384 bytes were initialized with data, but it was unclear as to what specifically the data is, as it's not a single value, and there is no discernable pattern.
    - The next 16384 bytes were initialized to "00" by the RAID controller. However, within this section there were a small set of randomly repeating hex values (80h, 40h, 04h, 20h, 01h) spread throughout this area.
    - The last 1818 bytes was initialized with data, but it was unclear as to specifically what the data was.

- The remaining portions of each drive had the wiped values on it, and were not changed during the RAID volume construction process.

RAID-DI-SC-09: This experiment was to see if the changes in the initial 64KB of each participating drive changes depending on the type of RAID implemented. Previously in BC-08, a RAID-5 was used; in this experiment a RAID-1 mirror was be constructed.

The system was setup with a Promise SX4000 controller, and 2 ATA drives configured to be a RAID-1 mirrored array. The drives were wiped before use, drive #1—"A1," and drive #2—"B2." The controller was allowed to construct the RAID-1 volume, and then each drive was examined using both DI-Tool-#2 as well as Norton DiskEdit.

Each drive had the first 64KB bytes initialized the same way as in RAID-DI-SC-08. Changing the type of RAID made no apparent impact.

RAID-DI-BC10: This experiment was to see if the number of drives participating in a RAID-5 affect how the first 64KB of the drives are initialized. In this case, three drives were used to construct the RAID-5, instead of four drives, and the changes made to the drives by the RAID controller were examined.

The system is configured using the Promise SX4000 controller, and 3 ATA drives are used to construct a RAID-5. Each drive was wiped with a specific value, drive #1—"A1," drive #2—"B2," and drive #3—"C3." The controller was used to construct a RAID-5, yielding a 120 GB RAID-5 volume. Each drive was examined using DI-Tool-#2, as well as Norton DiskEdit.

- Each drive had the first 64KB bytes initialized the same way as in RAID-DI-SC-08. Changing the number of participating drives made no apparent impact.

RAID-DI-SC-11: The standard configuration for RAID-5 is set for a 64KB stripe size. In this experiment, the stripe size was changed to 16KB to see what impact it would have on how participating drives are initialized by the RAID controller.

The system is configured using the Promise SX4000 controller, and 4 ATA drives are used to construct a RAID-5. Each drive was wiped with a specific value, drive #1—"A1," drive #2—"B2," drive #3—"C3," and drive #4—"D4." The controller was used to construct a RAID-5, yielding a 120 GB RAID-5 volume, but the stripe size was changed from 64KB to 16KB. Each drive was examined using DI-Tool-#2, as well as Norton DiskEdit.

- Each drive had the first 64KB bytes initialized in the same way as in RAID-DI-SC-08. Changing the block size made no apparent impact.

3.2.2 Findings

On closer examination, there were clearly situations where imaging the RAID volume did not provide completeness for a variety of reasons. However, the common thread in all of the boundary cases is that disk imaging was done through a hardware controller. As previously stated, if hard drives are individually imaged a complete and accurate image of the whole RAID will be obtained. If an image of a RAID volume is obtained through the controller, there are varieties of situations where the image will not be complete.

3.3 Impact of RAID on Disk Imaging

This section addresses the impact of findings on the activities of forensic investigators.

3.3.1 Common Usage/Imaging

The overall findings on the impact of RAID on disk imaging software suggest that in most cases RAID has no impact on the disk imaging process. In the cases where a disk imaging tool is affected, it was not due to a tool flaw, but in how the RAID hardware controller operates.

In every case, regardless of the type of RAID (Mirror, or Striping/Parity), drive type (SCSI, ATA), or controller type (Adaptec, Promise), the available RAID volume was completely and accurately imaged. Hashes of the overall RAID volume matched both before the imaging, and after restoration. Additionally, all created file system partitions on the RAID matched before being imaged and after being restored.

Furthermore, if all drives in a given RAID are imaged separately using forensically sound practices, then in all cases, RAID had no impact on disk imaging process. All information on each physical drive was captured, including all RAID volumes, as well as additional data that was not visible through the hardware RAID controller.

The only case where disk imaging was affected was when the RAID volume was imaged intact through the hardware controller. In this case, only the data within the constructed RAID volume was captured, and did not include all of the data on the participating drives (outside of the defined volume). It is important to note that this discrepancy was not due to a flaw in the imaging tools, but mainly due to two properties of the controller. First, RAID uses a small amount of overhead to keep track of the participating drives. This data is not accessible through the controller, but only visible if the drives are imaged separately. Second, since the drives participating in a given RAID could be of different sizes, the RAID controller provides some flexibility in how the size of the RAID is configured. The size of the partition could range in size from a small percentage of the drive, to a maximum size contingent on the smallest disk participating in the array (essentially the smallest participating drive limits how large the RAID volume can be).

3.3.2 Special Cases

In the course of these experiments, there were some factors which came to light that may be of use to investigators who encounter RAID systems during an examination. First, due to how RAID controllers work, there are ways to hide partitions on a given array. In test case RAID-DI-SC-05, a small 5 GB partition was hidden on a RAID-1 mirror. If the examiners only had access to the constructed array (through the RAID controller), there would be no way of identifying the hidden array or capturing the data within that partition.

Second, investigators need to be aware that the hashes of drives participating in RAID may not be the same. In the case of a RAID-5, where information is spread in parallel over each drive, this is clear, as each drive has different data and so the resulting hashes are different. However, in the case of RAID-1 mirrors, this holds true as well, due to information that the controller writes on the drive, and residual information, hashes of mirrored drives will vary as well. The hashes of the overall RAID mirror partition will not match any of the individual drive hashes. However, if only the data partitions are hashed, and not the whole physical drive, the hashes will be equal provided each partition was wiped before use.

Third, although RAID is implemented from a variety of manufacturers, there is no guarantee that they are compatible with each other. If an investigator encounters a RAID in the field, they need to take special care in recording all data about the hardware and software that is used to construct the array. If the RAID is unable to be reconstructed, the investigator has very little useful information that can be derived from an unconstructed RAID array. Although in the case of a RAID-1 mirror, since the data is directly duplicated, each drive is complete by itself. However, with the RAID-5 or RAID-0, data is distributed across multiple drives, and does not provide much usable data that is coherent.

4.0 Summary and Recommended Changes to Disk Imaging Specifications.

In summary, RAID generally did not impact disk imaging under most circumstances. However, there were some recommendations that we felt should be considered that might expand and make testing the Disk Imaging more complete.

Additionally, in during this investigation some particular aspects of RAID that forensic examiners should be aware of were identified.

4.1 Recommended Changes to Disk Imaging Specification

The research suggests that in almost all cases, disk imaging tools are not affected by RAID. Where there is an interaction, such as is the case where a hardware RAID volume is imaged intact, the disk imaging software works correctly. However, only the visible RAID volume is imaged, which may not include all the potential data on the participating hard drives. This is not a limitation of the disk imaging software, but properties of the hardware RAID. Overall, no changes are needed for the disk imaging specifications.

One final note; it may be wise to include hardware RAID within the actual testing environment. This would provide additional support to the disk imaging validation process, and potentially could provide forensic examiners additional insight on how RAID functions within a given situation and setting.

The initial specifications developed by NIST on disk imaging did not consider RAID. As this disk technology has become more prevalent, and encountered by law enforcement more frequently, research was done to determine if it affected tools used to image target media. Although a wide variety of RAID related factors were examined, such as RAID type, hardware and software RAID, and different types of drives (ATA vs. SCSI), RAID had virtually no impact on the disk imaging process.

As the research identified, the interaction between RAID and disk imaging can generally be broken down into three cases. The first being RAID constructed using software; second, RAID systems that were hardware based, but the participating drives were imaged separately using forensically sound procedures, and third, hardware RAID in which was imaged intact, with all activity taking place through the mediation of the RAID controller.

In the first case, where the RAID is constructed using software, disk imaging was not impacted at all. Essentially, since all drives are visible to the BIOS, and consequently the operating system, the imaging tools were able to image the drives directly. This was verified by using Linux to create a software RAID-1, and a software RAID-5, and then imaging the drives, and verifying the images. Additionally, Windows 2000 was used to create a dynamic disk drive utilizing 2 drives, and was imaged without difficulty. From the imaging tool point of view, since it can access the drives individually, it can copy bit-bit, and it does not matter what particular data or format of data is on the drives.

For the second case, where hardware RAID is used, but the drives are imaged separately, again the disk imaging tools were not affected. As in the first case, since each drive is imaged separately outside of the RAID controller, the imaging tool has access to entire drive and can get an accurate bit-bit image.

In the last case, where a hardware RAID volume is imaged through an intact controller, disk imaging was affected. However, this interaction was not due to a defect or problem with the disk imaging tools, but due to a property of the RAID controller. Since the

RAID is built using hardware, the controller only allows the constructed RAID volume to be visible to the computer BIOS. A volume, by definition, may or may not include all of the underlying disk space, and specifically for a RAID, it will not include all the available data on any of the participating disks. Since RAID volumes have additional overhead to keep track of the drives participating in an array, these data areas are not available to the usable RAID volume and decrease the overall size slightly.

4.2 Points of Interest to Forensic Examiners/Field Technicians

Throughout the investigation between RAID and disk imaging, there were a few points of interest that may be of use to forensic investigators who encounter RAID in the field. First, that hardware RAID can be used to hide data partitions. Although this takes moving the hard drive to a non-RAID controller, it could easily be done, and be difficult to detect from a forensic perspective if the RAID volume is only imaged through the controller. However, if standard forensic practice is used to image each drive separately, then the data is visible as with any other data partition.

Second, if a mirrored RAID is encountered, the hashes between all participating drives and the RAID volume may not match. Even though the common perception is that mirrored drives are identical, in reality, at the bit level, they are not—the data between the drives is identical, but not the whole drive. The RAID controller places unique information on the drives outside of the data area that can change the hash. Additionally, if the drives were not individually wiped before being used in the RAID, a bit-bit hash may include residual information that is not present on all drives. Therefore, if each drive is hashed, it will generate a unique value, and be different than the overall RAID volume.

Third, although most investigators are very aware of this point, hardware RAID controllers may not implement RAID types the same, and consequently will not be compatible with each other. It is critically important to have the information to reconstruct the array after it is acquired. This includes information at the BIOS level such as stripe size, number of participating drives and their order, as well as information on the hardware controller.

5.0 Appendix A: Test Case Template

Test Case Identification	RAID-DI-TC-01
RAID Implementation	Hardware or Software
RAID Type	0, 1, 5
Manufacturer/Cart Type	Promise SX4000
# Participating Drives	2, 4
Setup and Configuration	Brief overview of setup and configuration.
Created Partition Information	Partitions Created and Size
Data Gathering	Brief overview of how the data was gathered, and what specific information was recorded.
Findings	Summary of Analysis/Findings
Comments	

6.0 Appendix B: RAID/Disk Imaging Tests (Common Usage)

6.1 RAID-DI-TC-01: DI-Tool-#2: [Promise SX4000 RAID-1, 2 ATA Drives]

Test Case Identification	RAID-DI-TC-01
RAID Implementation	Hardware
RAID Type	Type 1
Manufacturer/Type	Promise SX4000
# Participating Drives	2 ATA
Overview	The main goal of this experiment is to see if there are any difficulties with the acquire disk imaging function in DI-Tool-#2 in detecting, reading and acquiring a 2 ATA (EIDE) drive hardware RAID-1 (mirror).
Setup and Configuration	• The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Promise SX4000 controller used. • 2 Barracuda 7200.7, 40 GB drives (Channel 1, 2). • RAID-1 (mirror) was setup utilizing two 40 GB ATA(EIDE) drives. • At bios prompt, RAID-1 configured to use both drives, and the maximum mirror size (40 GBs). • Partition magic 6 used at dos prompt to create 3 partitions. • PM6 was configured to look for bad sectors on each partition and then wipe the sector with null values.
Created Partition Information	Partition #1: FAT, 100 MB Partition #2: FAT-32, 100 MB Partition #3: NTFS, 100 MB
Data Gathering	• DI-Tool-#2 en.exe used from CD boot disk to acquire the RAID-1 mirror through the hardware controller. o RAID-1 mirror was hashed before and after acquisition. o RAID-1 mirror was restored to new drive, and hashed. • Each (2) participating drive was separately mounted and acquired with DI-Tool-#2 en.exe. • Each drive hashed before and after acquisition.
Findings	• The RAID-1 mirror was acquired completely and accurately by using DI-Tool-#2 en.exe. Hashes matched during and after the acquisition step and the restored image matched in hash value. No errors were encountered during acquisition. • Each individual drive hashed the same before and after acquisition. However, they had different hash values between themselves and the RAID-1 mirror hash value. • Each partition on the individual drives hashed the same as the constructed RAID-1 mirror partition. o #1 FAT, hashed the same on both individual drives, and the RAID-1. o #2 FAT-32, hashed the same on both individual drives, and the RAID-1. o #3 NTFS, hashed the same on both individual drives, and the RAID-1.
Comments	

6.2 RAID-DI-TC-02: DI-Tool-#2: [Promise SX4000 RAID-5, 4 ATA Drives]

Test Case Identification	RAID-DI-TC-02
RAID Implementation	Hardware
RAID Type	Type 5
Manufacturer/Type	Promise SX4000
# Participating Drives	4 ATA
Overview	The main goal of this experiment is to see if there is any difficulties with the acquire disk imaging function in DI-Tool-#2 in detecting, reading and acquiring a 4 ATA (EIDE) drive hardware based RAID-5 (Stripe/Parity).
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Promise SX4000 controller used. • 4 Barracuda 7200, 40 GB ATA(EIDE) drives (channels 1, 2, 3, 4). • At the BIOS prompt, RAID-5 (Stripe/Parity) was setup utilizing all drives, yielding a 120-GB RAID-5 partition. • Partition magic 6 used at dos prompt to create 3 partitions. It also scanned each partition for bad sectors, and wiped each sector during the process.
Created Partition Information	Partition #1: FAT, 100 MB Partition #2: FAT-32, 100 MB Partition #3: NTFS, 100 MB
Data Gathering	• DI-Tool-#2 en.exe used from the CD Boot disk to acquire the RAID-5 (Stripe/Parity) partition. o RAID-5 hashed before and after acquisition. • Each participating drive was acquired separately, hashed before and after being restored. • Drives were restored on wiped drives, and re-inserted to RAID, and hashes were recalculated. o Each drive was wiped. o Each individual image was restored to a different 40-GB drive. o Drives re-inserted into RAID array, and RAID brought online, rehashed.
Findings	The RAID-5 (Stripe/Parity) partition was acquired by DI-Tool-#2 en.exe, without any difficulties, and the restored image hashed correctly. Each drive was also imaged, and was correctly restored and hashed as well. • Partition #1 FAT, hashed the same as the original and on the restored drives. • Partition #2 FAT-32, hashed the same as the original and on the restored drives. • Partition #3 NTFS, hashed the same as on the original and on the restored drives. Additionally, after creating images, each target drive was wiped, and the images were restored onto a different drive (#1 restored to #2, #2 to #3, #3 to #4, and #4 to #1). The drives were inserted back into the RAID-5 in the appropriate order. The overall RAID-5 partition hashed the same as before, as well as each partition within the RAID-5.
Comments	

6.3 RAID-DI-TC-03: DI-Tool-#2: [Adaptec 2110S RAID-1, 2 SCSI Drives]

Test Case Identification	RAID-DI-TC-03
RAID Implementation	Type 1
RAID Type	Hardware
Manufacturer/Type	Adaptec 2110S
# Participating Drives	2 SCSI
Overview	The main goal of this experiment, as to see if DI-Tool-#2 has any difficulties with the acquire disk imaging function detecting, reading, and imaging a SCSI 2 drive hardware based RAID-1 (mirror).
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Adaptec 2110S controller used. • 2 Seagate Cheetah 320, 18.4 GB SCSI drives, ID (1, 2). • At the BIOS prompt, RAID-1 was setup utilizing all drives, yielding an 18.4 GB RAID-1 partition. • Partition magic 6 used at dos prompt to create 3 partitions. It also scanned each partition for bad sectors, and wiped each sector during the process.
Created Partition Information	Partition #1: FAT, 100 MB Partition #2: FAT-32, 100 MB Partition #3: NTFS, 100 MB
Data Gathering	• DI-Tool-#2 en.exe used from the CD boot disk to acquire the RAID-1 (mirror) partition. • RAID-1 hashed before and after imaging. • RAID-1 hashed after restoring back to SCSI RAID-1.
Findings	The RAID-1 partition was imaged without difficulty. • The original RAID-1 partition hashed the same before and after being acquired. • The restored partition also matched with the original hashes. • Partition #1 FAT, hashed the same as the original and on the restored drives. • Partition #2 FAT-32, hashed the same as the original and on the restored drives. • Partition #3 NTFS, hashed the same as on the original and on the restored drives.
Comments	

6.4 RAID-DI-TC-04: DI-Tool-#2: [Adaptec 2100S RAID-5, 4 SCSI Drives]

Test Case Identification	RAID-DI-TC-04
RAID Implementation	Type 5
RAID Type	Hardware
Manufacturer/Type	Adaptec 2110S
# Participating Drives	4 SCSI
Overview	The main goal of this experiment is to see if there is any difficulties with the acquire disk imaging function in DI-Tool-#2 in detecting, reading and acquiring a 4 SCSI drive hardware based RAID-5 (Stripe/Parity) partition.
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Adaptec 2100S controller used. • 4 Seagate Cheetah 320, 18.4 GB SCSI drives, ID (1, 2, 3, 4). • At the BIOS prompt, RAID-5 (Stripe/Parity) was setup utilizing all drives, yielding a 55.2 GB RAID-5 partition. • Partition magic 6 used at dos prompt to create 3 partitions. It also scanned each partition for bad sectors, and wiped each sector during the process.
Created Partition Information	Partition #1: FAT, 100 MB Partition #2: FAT-32, 100 MB Partition #3: NTFS, 100 MB
Data Gathering	DI-Tool-#2 en.exe used from the CD Boot disk to acquire the RAID-5 (Stripe/Parity) partition. • RAID-5 hashed before, and after acquisition. • All partitions hashed before, and after acquisition. • RAID-5 partition restored, and hashed.
Findings	The RAID-5 (Stripe/Parity) partition was acquired without any difficulties, and the restored image hashed correctly. • Partition #1 FAT, hashed the same as the original and on the restored drives. • Partition #2 FAT-32, hashed the same as the original and on the restored drives. • Partition #3 NTFS, hashed the same as the original and on the restored drives.
Comments	

6.5 RAID-DI-TC-05: DI-Tool-#1: [Promise SX4000 RAID-1, 2 ATA Drives]

Test Case Identification	RAID-DI-TC-05
RAID Implementation	Type 1
RAID Type	Hardware
Manufacturer/Type	Promise SX4000
# Participating Drives	2 ATA
Overview	The main goal of this experiment is to see if there are any difficulties with the acquire disk imaging function in DI-Tool-#1 in detecting, reading and acquiring a 2 ATA (EIDE) drive hardware RAID-1 (mirror).
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Promise SX4000 controller used. • 2 Barracuda 7200.7, 40 GB drives (Channel 1, 2). • RAID-1 (mirror) was setup utilizing two 40 GB ATA(EIDE) drives. • At bios prompt, RAID-1 configured to use both drives, and the maximum mirror size (40 GBs). • Partition magic 6 used at dos prompt to create 3 partitions.
Created Partition Information	Partition #1: FAT, 100 MB Partition #2: FAT-32, 100 MB Partition #3: NTFS, 100 MB PM6 was configured to look for bad sectors on each partition, and wiped each sector while doing so.
Data Gathering	• DI-Tool-#1 used from CFTT CD boot disk to acquire the RAID-1 mirror through the hardware controller. • RAID-1 mirror was hashed before and after acquisition. • RAID-1 mirror was restored to new drive, and hashed. • Each (2) participating drive was separately mounted and acquired with DI-Tool-#1. • Each drive hashed before and after acquisition.
Findings	• The RAID-1 mirror was acquired completely and accurately through DI-Tool-#1. Hashes matched before and after the acquisition step and the restored image matched in hash value. No errors were encountered during acquisition. • Each individual drive hashed the same before and after acquisition. However, they had different hash values between themselves and the RAID-1 mirror hash value. • Each partition on the individual drives hashed the same as the constructed RAID-1 mirror partition. o #1 FAT, hashed the same on both individual drives, and the RAID-1. o #2 FAT-32, hashed the same on both individual drives, and the RAID-1. o #3 NTFS, hashed the same on both individual drives, and the RAID-1.
Comments	

6.6 RAID-DI-TC-06: DI-Tool-#1: [Promise SX4000 RAID-5, 4 ATA/EIDE Drives]

Test Case Identification	RAID-DI-TC-06
RAID Implementation	Type 5
RAID Type	Hardware
Manufacturer/Type	Promise SX4000
# Participating Drives	4 ATA
Overview	The main goal of this experiment is to see if there is any difficulties with the acquire disk imaging function in DI-Tool-#1 in detecting, reading and acquiring a 4 ATA (EIDE) drive RAID-5 (Stripe/Parity) partition.
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Promise SX4000 controller used. • 4 Barracuda 7200, 40 GB ATA(EIDE) drives (channels 1, 2, 3, 4). • At the BIOS prompt, RAID-5 (Stripe/Parity) was setup utilizing all drives, yielding a 120-GB RAID-5 partition. • Partition magic 6 used at dos prompt to create 3 partitions. It also scanned each partition for bad sectors, and wiped each sector during the process
Created Partition Information	Partition #1: FAT, 100 MB Partition #2: FAT-32, 100 MB Partition #3: NTFS, 100 MB
Data Gathering	• DI-Tool-#1 used from the CFTT CD Boot disk to acquire the RAID-5 (Stripe/Parity) partition. • RAID-5 hashed before and after acquisition. • Each participating drive was acquired separately, hashed before and after being restored. • Drives were restored on wiped drives, and re-inserted into the RAID controller, and hashes were recalculated. • Each drive was wiped. • Each individual image was restored to a different 40-GB drive. • Drives re-inserted into RAID array, RAID brought online, and all partitions hashed.
Findings	The RAID-5 (Stripe/Parity) partition was acquired without any difficulties, and the restored image hashed correctly. Each drive was also imaged, and was correctly restored and hashed as well. • Partition #1 FAT, hashed the same as the original and on the restored drives. • Partition #2 FAT-32, hashed the same as the original and on the restored drives. • Partition #3 NTFS, hashed the same as on the original and on the restored drives.
Comments	After creating images, each target drive was wiped, and the images were restored onto a different drive (#1 restored to #2, #2 to #3, #3 to #4, and #4 to #1). The drives were inserted back into the RAID-5 in the correct order. The overall RAID-5 partition hashed the same as before, as well as each partition within the RAID-5.

6.7 RAID-DI-TC-07: DI-Tool-#1: [Adaptec 2100S RAID-1, 2 SCSI Drives]

Test Case Identification	RAID-DI-TC-07
RAID Implementation	Type 1
RAID Type	Hardware
Manufacturer/Type	Adaptec 2110S
# Participating Drives	2 SCSI
Overview	The main goal of this experiment is to see if there are any difficulties with the acquire disk imaging function in DI-Tool-#1 in detecting, reading and acquiring a 2 SCSI drive RAID-1 mirror.
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Adaptec 2100S controller used. • 2 Seagate Cheetah 320, 18.4 GB SCSI drives, ID (1, 2). • At the BIOS prompt, RAID-1 mirror was setup utilizing two drives, yielding an 18.4 GB RAID-5 partition. • Partition magic 6 used at dos prompt to create 3 partitions.
Created Partition Information	PM6 was configured to look for bad sectors on each partition, and wiped each sector while doing so. Partition #1: FAT, 100 MB Partition #2: FAT-32, 100 MB Partition #3: NTFS, 100 MB
Data Gathering	• DI-Tool-#1 used from CFTT CD boot disk to acquire the RAID-1 mirror through the hardware controller. o RAID-1 mirror was hashed before and after acquisition. o RAID-1 mirror was restored to new drive, and hashed. • Each (2) participating drive was separately mounted and acquired with DI-Tool-#1. o Each drive hashed before and after acquisition.
Findings	• The RAID-1 mirror was accurately and completely acquired through DI-Tool-#1. Hashes matched before and after the acquisition step and the restored image matched in hash value. No errors were encountered during acquisition. • Each individual drive hashed the same before and after acquisition. However, they had different hash values between themselves and the RAID-1 mirror hash value. • Each partition on the individual drives hashed the same as the constructed RAID-1 mirror partition. o #1 FAT, hashed the same on both individual drives, and the RAID-1. o #2 FAT-32, hashed the same on both individual drives, and the RAID-1. o #3 NTFS, hashed the same on both individual drives, and the RAID-1.
Comments	

6.8 RAID-DI-TC-08: DI-Tool-#1: [Adaptec 2110S RAID-5, 4 SCSI Drives]

Test Case Identification	RAID-DI-TC-08
RAID Implementation	Type 5
RAID Type	Hardware
Manufacturer/Type	Adaptec 2110S
# Participating Drives	4 SCSI
Overview	The main goal of this experiment is to see if there is any difficulties with the acquire disk imaging function in DI-Tool-#1 in detecting, reading and acquiring a 4 SCSI drive RAID-5 (Stripe/Parity) partition.
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Adaptec 2110S controller used. • 4 Seagate Cheetah 320, 18.4 GB SCSI drives, ID (1, 2, 3, 4). • At the BIOS prompt, RAID-5 (Stripe/Parity) was setup utilizing all drives, yielding a 52 GB RAID-5 partition. • Partition magic 6 used at dos prompt to create 3 partitions. It also scanned each partition for bad sectors, and wiped each sector during the process.
Created Partition Information	Partition #1: FAT, 100 MB Partition #2: FAT-32, 100 MB Partition #3: NTFS, 100 MB
Data Gathering	• DI-Tool-#1 used from CFTT CD Boot disk to acquire the RAID-5 (Stripe/Parity) partition. o RAID-5 hashed before and after acquisition. o All individual partitions (#1, #2) hashed before, and after acquisition. o The original RAID-5 erased, all drives in the array moved +1 in order, the image restored onto the new RAID-5 array. The RAID-5 partition hashed correctly after restoration.
Findings	The RAID-5 (Stripe/Parity) partition was acquired without any difficulties, and the restored image hashed correctly. • Partition #1 FAT, hashed the same as the original and on the restored drives. • Partition #2 FAT-32, hashed the same as the original and on the restored drives. • Partition #3 NTFS, hashed the same on the original and on the restored drive.
Comments	

6.9 RAID-DI-TC-09: DI-Tool-#3 Imaging: [Promise SX4000 RAID-1, 2 ATA Drives]

Test Case Identification	RAID-DI-TC-09
RAID Implementation	Type 1
RAID Type	Hardware
Manufacturer/Type	Promise SX4000
# Participating Drives	2 ATA
Overview	The main goal of this experiment is to see if the DI-Tool-#3 can read, and image a 2 ATA (EIDE) drive hardware RAID-1 (mirror).
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Promise SX4000 controller used. • 2 Barracuda 7200.7, 40 GB drives (Channel 1, 2). • RAID-1 (mirror) was setup utilizing two 40 GB ATA(EIDE) drives. • At bios prompt, RAID-1 configured to use both drives, and the maximum mirror size (40 GBs). • Partition magic 6 used at dos prompt to create 3 partitions.
Created Partition Information	Partition #1: FAT, 100 MB Partition #2: FAT-32, 100 MB Partition #3: NTFS, 100 MB PM6 was configured to look for bad sectors on each partition and then wipe the sector with null values.
Data Gathering	• DI-Tool-#3 was used to image the RAID-1 mirror through the hardware controller. o RAID-1 mirror was hashed before and after acquisition. o RAID-1 mirror was restored to new drive, and hashed. • Each (2) participating drive was separately mounted and imaged with DI-Tool-#3, and each drive was hashed before and after imaging.
Findings	• The RAID-1 mirror was imaged completely and accurately by using DI-Tool-#3. Hashes matched during and after the imaging step and the restored image matched in hash value. No errors were encountered during imaging. • Each individual drive hashed the same before and after imaging. However, they had different hash values between themselves and the RAID-1 mirror hash value. • Each partition on the individual drives hashed the same as the constructed RAID-1 mirror partition. o #1 FAT, hashed the same on both individual drives, and the RAID-1. o #2 FAT-32, hashed the same on both individual drives, and the RAID-1. o #3 NTFS, hashed the same on both individual drives, and the RAID-1.
Comments	

6.10 RAID-DI-TC-10: DI-Tool-#3 Imaging:[Promise SX4000 RAID-5, 4 ATA Drives]

Test Case Identification	RAID-DI-TC-10
RAID Implementation	Type 5
RAID Type	Hardware
Manufacturer/Type	Promise SX4000
# Participating Drives	4 ATA
Overview	The main goal of this experiment is to see if DI-Tool-#3 has any difficulties reading and imaging a 4 ATA (EIDE) drive hardware based RAID-5 (Stripe/Parity).
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Promise SX4000 controller used. • 4 Barracuda 7200, 40 GB ATA(EIDE) drives (channels 1, 2, 3, 4). • At the BIOS prompt, RAID-5 (Stripe/Parity) was setup utilizing all drives, yielding a 120-GB RAID-5 partition. • Partition magic 6 used at dos prompt to create 3 partitions. It also scanned each partition for bad sectors, and wiped each sector during the process.
Created Partition Information	Partition #1: FAT, 100 MB Partition #2: FAT-32, 100 MB Partition #3: NTFS, 100 MB
Data Gathering	• DI-Tool-#3 was used to image the array, and was hashed before and after the imaging. • Each participating drive was imaged separately, hashed before and after being restored. • Drives were restored on wiped drives, and re-inserted to RAID, and hashes were recalculated. o Each drive was wiped. o Each individual image was restored to a different 40-GB drive. o Drives re-inserted into RAID array, and RAID brought online, rehashed.
Findings	The RAID-5 (Stripe/Parity) partition was imaged through DI-Tool-#3 without any problems, and the restored imaged hashed correctly. Additionally, each individual drive was also imaged and restored correctly as well. • Partition #1 FAT, hashed the same as the original and on the restored drives. • Partition #2 FAT-32, hashed the same as the original and on the restored drives. • Partition #3 NTFS, hashed the same as on the original and on the restored drives.
Comments	After creating images, each target drive was wiped, and the images were restored onto a different drive (#1 restored to #2, #2 to #3, #3 to #4, and #4 to #1). The drives were inserted back into the RAID-5 in the appropriate order. The overall RAID-5 partition hashed the same as before, as well as each partition within the RAID-5.

6.11 RAID-DI-TC-11: DI-Tool-#3 Imaging: [Adaptec 2110S RAID-1, 2 SCSI Drives]

Test Case Identification	RAID-DI-TC-11
RAID Implementation	Type 1
RAID Type	Hardware
Manufacturer/Type	Adaptec 2110S
# Participating Drives	2 SCSI
Overview	The main goal of this experiment is to see if DI-Tool-#3 can read and image a SCSI 2 drive hardware based RAID-1 (mirror).
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Adaptec 2110S controller used. • 2 Seagate Cheetah 320, 18.4 GB SCSI drives, ID (1, 2). • At the BIOS prompt, RAID-1 was setup utilizing all drives, yielding an 18.4 GB RAID-1 partition. • Partition magic 6 used at dos prompt to create 3 partitions. It also scanned each partition for bad sectors, and wiped each sector during the process.
Created Partition Information	Partition #1: FAT, 100 MB Partition #2: FAT-32, 100 MB Partition #3: NTFS, 100 MB
Data Gathering	• DI-Tool-#3 was used to image the RAID-1 (mirror) partition. • RAID-1 hashed before and after imaging. • RAID-1 hashed after restoring back to SCSI RAID-1
Findings	The RAID-1 partition was imaged without difficulty. • The original RAID-1 partition hashed the same before and after being imaged. • The restored partition also matched with the original hashes. • Partition #1 FAT, hashed the same as the original and on the restored drives. • Partition #2 FAT-32, hashed the same as the original and on the restored drives. • Partition #3 NTFS, hashed the same as on the original and on the restored drives.
Comments	

6.12 RAID-DI-TC-12: DI-Tool-#3 Imaging: [Adaptec 2100S RAID-5, 4 SCSI Drives]

Test Case Identification	RAID-DI-TC-12
RAID Implementation	Type 5
RAID Type	Hardware
Manufacturer/Type	Adaptec 2110S
# Participating Drives	4 SCSI
Overview	The main goal of this experiment is to see if there are any difficulties in DI-Tool-#3 reading and imaging a 4 drive SCSI hardware based RAID-5 (stripe/parity) partition.
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Adaptec 2100S controller used. • 4 Seagate Cheetah 320, 18.4 GB SCSI drives, ID (1, 2, 3, 4). • At the BIOS prompt, RAID-5 (Stripe/Parity) was setup utilizing all drives, yielding a 55.2 GB RAID-5 partition. • Partition magic 6 used at dos prompt to create 3 partitions. It also scanned each partition for bad sectors, and wiped each sector during the process
Created Partition Information	Partition #1: FAT, 100 MB Partition #2: FAT-32, 100 MB Partition #3: NTFS, 100 MB
Data Gathering	DI-Tool-#3 was used to image the RAID-5 (stripe/parity) partition • RAID-5 hashed before, and after acquisition. • All partitions hashed before, and after acquisition. • RAID-5 partition restored, and hashed.
Findings	The RAID-5 (Stripe/Parity) partition was imaged without any difficulties, and the restored image hashed correctly. • Partition #1 FAT, hashed the same as the original and on the restored drives. • Partition #2 FAT-32, hashed the same as the original and on the restored drives. • Partition #3 NTFS, hashed the same as the original and on the restored drives.
Comments	

7.0 Appendix C: RAID/Disk Imaging Tests (Special Cases)

7.1 RAID-DI-SC-01: Size Differences between RAID-1 Volume and Individual ATA/EIDE Drives

Test Case Identification	RAID-DI-SC-01
RAID Implementation	Type 1
RAID Type	Hardware
Manufacturer/Type	Promise SX4000
# Participating Drives	2 ATA
Overview	There appeared to be a difference between the size of the constructed RAID-1 mirror image, and the actual size on each drive in the array, even when each drive was identical and the RAID-1 array was maximized. In this experiment, each drive is examined for its size relative to the RAID-1 mirror array.
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Promise SX4000 controller used. • 2 Barracuda 7200.7, 40 GB drives (Channels 1, 2). • RAID-1 (mirror) was setup utilizing both 40 GB drives. • At bios prompt, RAID-1 configured to use both drives, and the maximum mirror size (40 GBs). • Partition magic 6 used at dos prompt to create 3 partitions.
Created Partition Information	PM6 was configured to look for bad sectors on each partition, and wiped each sector while doing so. Partition #1: FAT, 100 MB Partition #2: FAT-32, 100 MB Partition #3: NTFS, 100 MB
Data Gathering	• The RAID-1 mirror sector size was recorded. • Each individual drive in the array had its size recorded.
Findings	Due to the overhead of the RAID-1 mirror, it is smaller than either of the participating drives. Additionally, if one of the participating drives is smaller than the other, this is the limiting factor for the mirror, and the RAID-1 size is correlated to the smallest disk size.
Comments	

7.2 RAID-DI-SC-02: Hash differences between RAID-1 Volume and Individual ATA/EIDE Drives

Test Case Identification	RAID-DI-SC-02
RAID Implementation	Type 1
RAID Type	Hardware
Manufacturer/Type	Promise SX4000
# Participating Drives	2 ATA
Overview	In earlier experiments, the individual drives used in a RAID-1 mirror hashed to different values. This experiment is looking directly at this, seeing if it's a result of residual information, or unique RAID information written the drives when the array is built or utilized. Note, this is only looking at the whole drive hash and not the partitions on each drive.
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Promise SX4000 controller used. • 2 Barracuda 7200.7, 40 GB ATA(EIDE) drives (Channels 1, 2). • RAID-1 (mirror) was setup utilizing both 40 GB drives. • At bios prompt, RAID-1 configured to use both drives, and the maximum mirror size (40 GBs). • Partition magic 6 used at dos prompt to create 3 partitions.
Created Partition Information	PM6 was configured to look for bad sectors on each partition, and wiped each sector while doing so. Partition #1: FAT, 100 MB Partition #2: FAT-32, 100 MB Partition #3: NTFS, 100 MB
Data Gathering	• The RAID-1 mirror was imaged using the DI-Tool-#2 Boot CD. • Each drive was imaged using the DI-Tool-#2 Boot CD. • Hashes were taken before and after imaging of the RAID-1, as well as both participating drives. • A separate disk wipe program (from our testing software) wrote FF to all sectors to both participating drives.
Findings	• Hardware RAID-1 mirrors when forensically examined may have hash value discrepancies. Although the partitions within each RAID-1 will hash the same, the overall drive hashes do not match. The reason for this depends on mainly three factors. • As with earlier experiments, there is a size difference between the constructed RAID-1 mirror, and each participating drive. Since hashing is a bit-bit process, size differences, even with null values, will change the resulting hash values. • The RAID controller places information on each participating drive to track drive number and status. Even if each drive were the same size, and bit-bit identical, there is unique RAID information that would cause the hashes to differ. • Forensic hashing of a disk is bit-bit, if the original drives were not wiped before use, the residual information in each drive would cause the hashing to differ. The RAID is similar to most file systems in that it is conservative, and only keeps written information accurate and is not concerned about residual information on the drives. Previous information that was present on the individual drives is not removed, and is only overwritten. When the drives are hashed individually, this residual information affects the hashing process, and produces differing hash values for each drive when compared to each other and to the constructed RAID-1 mirror[8].
Comments	

[8] For Adaptec controllers, if the hardware RAID controller is allowed to fully construct the RAID, it appears to wipe the partition as it constructs it.

7.3 RAID-DI-SC-03: Hardware RAID-1 Drive Bias for Residual Data

Test Case Identification	RAID-DI-SC-03
RAID Implementation	Type 1
RAID Type	Hardware
Manufacturer/Type	Promise SX4000
# Participating Drives	2 ATA
Overview	In the RAID-1 documentation, reading from a mirrored RAID has the advantage of being able to read data from both drives, bypassing single drive throughput limitations. However, within the realm of a forensic examination, this could impact what type of residual data recovered from a drive. When a bit-bit image is performed on a RAID drive, depending on how the reads are performed, the returned residual information may be from any drives participating in the RAID-1 or biased from a particular drive.
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Promise SX4000 controller used. • 2 Barracuda 7200.7, 40 GB ATA(EIDE) drives (Channels 1, 2). • Drive #1: Wiped with the value "11". • Drive #2: Wiped with the value "FF". • RAID-1 (mirror) was setup utilizing both 40 GB drives. • At bios prompt, RAID-1 configured to use both drives, and the maximum mirror size (40 GBs).
Created Partition Information	Partition Magic used to create two, 1 GB partitions (Fat 32, and NTFS), with each partition searched for bad sectors and wiped.
Data Gathering	• The RAID-1 Mirror was imaged using DI-Tool-#1, DI-Tool-#2, and DI-Tool-#3, and hashed before and after acquisition. • Each Drive was imaged using DI-Tool-#1, DI-Tool-#2, and DI-Tool-#3, and hashed before and after imaging. • Norton DiskEdit used to view the RAID-1 mirror directly. • Participating drives #1 and #2 were physically swapped in the RAID-1 Array, and the RAID-1 was re-acquired. Norton DiskEdit was used to view the reconstructed RAID-1 directly.
Findings	There is a drive read bias for the Promise RAID card when it is configured for mirroring. For this controller, it appears to be dependent on which channel the drives are located on. If drives are located on channels 1 and 2, then the preferred channel is 2, and the residual data comes from the drive connected to that channel. • The partitions between the RAID-1 mirror, and each individual drive hashed to the same value. • The unallocated space on the RAID-1 volume for all sectors was "FF". All residual information on the actual RAID-1 mirror was from drive #2. • The RAID-1 was examined by using Norton DiskEdit. When unallocated sectors were directly viewed on the RAID-1, the only value found was "FF"—again residual information only from drive #2. • Drive #1 and #2 were physically swapped, and the RAID-1 was again examined with Norton DiskEdit. This time, the unallocated space only showed the values of "11", which was from drive #1, which at this time is located on channel 2.
Comments	

7.4 RAID-DI-SC-04: Hardware RAID-5 Data and Parity Distribution

Test Case Identification	RAID-DI-SC-04
RAID Implementation	Type 5
RAID Type	Hardware
Manufacturer/Type	Promise SX4000
# Participating Drives	4 ATA
Overview	RAID-5 (Stripe/Parity) takes a group of drives and distributes information across all of them. To reduce data loss, it also writes a parity block as well, which can be used to reconstruct missing data or a complete array if no more than a single drive is damaged. This experiment is to better understand what data is transferred from each drive during RAID-5 reads; if there is any special bias on any of the drives in the constructed RAID.
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Promise SX4000 controller used. • 4 Barracuda 7200, 40 GB ATA(EIDE) drives (Channels 1, 2, 3, 4). o Each drive was completely wiped before use with a special value. The drive #1 was wiped with the value "11". The second drive was wiped with "22", the third "33", and the fourth "44". • At the BIOS prompt, RAID-5 (Stripe/Parity) was setup utilizing all drives, yielding a 120 GB RAID-5 partition. • Partition magic 6 used at dos prompt to create 3 partitions, but the options to wipe the partition before use was disabled (normally the partition is checked for bad sectors, and wiped during the process.
Created Partition Information	Partition #1: FAT, 100 MB Partition #2: FAT-32, 100 MB Partition #3: NTFS, 100 MB
Data Gathering	• The RAID-5 partition was imaged using DI-Tool-#2, and hashed before and after acquisition. • Norton DiskEdit was used to view the RAID-5 partition, as well as the unallocated and non-partitioned space on the RAID-5 drive.
Findings	• The RAID-5 partition, and unallocated disk space operated as the specifications state, with data being spread across all the drives, and a rotating parity bit. • When the unallocated space was examined, both with Norton DiskEdit, and with DI-Tool-#2, the pattern of data showed out of every 4 blocks written, one was a parity block. If you followed the residual data (11, 22, 33, 44), and plot it out, the following pattern appears. The (P) is where a given block was not visible, and coincides with a rotating parity block. o (P) 22 33 44 o 11 22 33 (P) o 11 22 (P) 44 o 11 (P) 33 44 o (P) 22 33 44 o 11 22 33 (P) • The data was in 64KB blocks, with some additional overhead, which fits the BIOS RAID-5 setup. • The rotating parity was determined through hardware and not just from information written to the hard drives. If the information for parity were contained only on the hard drives, the disk editor would have shown data other than 11, 22, 33 or 44. • The original RAID-5 image was restored and hashed the same as the original RAID volume.
Comments	

7.5 RAID-DI-SC-05: Hiding Partitions within a RAID-1

Test Case Identification	RAID-DI-SC-05
RAID Implementation	Type 1
RAID Type	Hardware
Manufacturer/Type	Promise SX4000
# Participating Drives	2 ATA
Overview	Using information gained from RAID-1 drive biasing, it is possible under certain circumstances to construct an array where data can be hidden within the RAID-1 array, and invisible to a forensic examiner.
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Promise SX4000 controller used. • 2 Barracuda 7200.7, 40 GB ATA(EIDE) drives (Channels 1, 2). • A single RAID-1 mirror was setup using only 20 GBs of the possible 40 GBs on the participating drives. • One of the drives in the RAID-1 mirror is moved to another machine, and a 5 GB FAT-32 partition is created at the end of the drive. This drive is subsequently accessed through Windows, and files placed on the 5 GB partition. • The drive is re-inserted into the RAID-1 array, and then files are now placed on the RAID-1 mirror within the windows environment.
Created Partition Information	
Data Gathering	• The RAID-1 mirror is imaged using DI-Tool-#2 boot CD, hashed before and after imaging. • The RAID-1 mirror is examined using Norton DiskEdit, and DI-Tool-#2.
Findings	• If the bias is known for the RAID controller, the drive with the “secret” partition can be placed on that controller (i.e. the one that does not provide residual information). The RAID controller itself will ignore the additional partition, and only mount the RAID volume. This in itself will prevent the hidden partition/information from being discovered. Even if this was not the case, since any residual information is from the non-hidden partitions, again this will not give any indication that there is any other data on the drive other than within the RAID. • It is important to note that if the drives are imaged using standard forensic practice, (i.e. individually acquired/imaged); all information will be present, including the hidden partition.
Comments	

7.6 RAID-DI-SC-06: Forensic Examination on Non-Reconstructed RAID-5 Array

Test Case Identification	RAID-DI-SC-06
RAID Implementation	Type 5
RAID Type	Hardware
Manufacturer/Type	Promise SX4000
# Participating Drives	4 ATA
Overview	It may be the case that investigators will not have access to all the drives in a given RAID-5 array. If a single drive is missing, the complete RAID-5 partition may be reconstructed using the partition information distributed throughout the remaining drives. However, if more than a single drive is damaged or not available, the investigator may be forced to look at the drives individually. Given that circumstance, it is important to know what data may be available to a examiner, and if it is useful.
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Promise SX4000 controller used. • 4 Barracuda 7200, 40 GB ATA(EIDE) drives (Channels 1, 2, 3, 4). • At the BIOS prompt, RAID-5 (Stripe/Parity) was setup utilizing all drives, yielding a 120 GB RAID-5 partition.
Created Partition Information	Partition magic 6 used at dos prompt to create 2 partitions, but the options to wipe the partition before use was disabled (normally the partition is checked for bad sectors, and wiped during the process. Partition #1: FAT-32, 100 MB Partition #2: NTFS, 100 MB Each partition had a variety of programs copied to it (.exe, .txt, etc…).
Data Gathering	• Each drive in the RAID-5 was acquired separately, hashed before and after the drive image was made. • Each drive was examined and searched for data residing on the drive without the RAID-5 being reconstructed. • DI-Tool-#2 was used to reconstruct the RAID-5 array to verify hashing between the original partition and the recreated RAID-5 partition using disk images.
Findings	It is possible to do a forensic examination of a RAID-5 system without reconstructing the actual RAID-5 partition. However, the data returned is very limited, and generally without any definable context. • There is limited (if any) meta-data about the partitions, files, or any other aspect of the file system. • Some readable information from text files was visible on a few of the drives, but it was difficult to identify what specific file it came from. • It may be possible to identify small files and bookmark them, since the RAID-5 writes are in sequential 64KB blocks. Without additional testing, it is difficult to determine how valuable this is without file system or partition information. • As verification, the whole RAID-5 was reconstructed and the hashes compared between the original partition, and the reconstructed one—they matched.

7.7 RAID-DI-SC-07: Sector Differences on SCSI RAID-5, between PM 6, DI-Tool-#1, DI-Tool-#2

Test Case Identification	RAID-DI-SC-07
RAID Implementation	Type 5
RAID Type	Hardware
Manufacturer/Type	Adaptec 2100S
# Participating Drives	4 SCSI
Overview	During one of the experiments, there was as discrepancy in the number of total sectors being imaged from the SCSI RAID-5. A new SCSI RAID-5 was constructed, and was examined by Partition Magic 6, DI-Tool-#2 and DI-Tool-#1 to see if there were differences and where.
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Adaptec 2100S controller used. • 4 Seagate Cheetah 320, 18.4 GB SCSI drives, ID (1, 2, 3, 4). • At the BIOS prompt, RAID-5 (Stripe/Parity) was setup utilizing all drives, yielding a 55.2 GB RAID-5 partition. • Partition magic 6 used at dos prompt to create 3 partitions. It also scanned each partition for bad sectors, and wiped each sector during the process.
Created Partition Information	Partition #1: FAT, 100 MB Partition #2: FAT-32, 100 MB Partition #3: NTFS, 100 MB
Data Gathering	• Each tool examined the RAID-5 partition for size and sector information. • DiskHash (CFTT Tool) used to calculate hash on RAID-5 partition.
Findings	• The two forensic tools (DI-Tool-#1, DI-Tool-#2) reported a total of 107526144 sectors of accessible data. Partition Magic 6 reported less available sectors, 107522982. This is a total of 3162 sectors difference. • DiskHash (CFTT Tool), reported a total of 107526144 sectors, the same as the forensic tools, but ran extremely slowly calculating the SHA-1 hash on the active RAID volume.
Comments	

7.8 RAID-DI-SC-08: RAID-5 Information Written During 4-Drive Volume Construction

Test Case Identification	RAID-DI-SC-08
RAID Implementation	Type 5
RAID Type	Hardware
Manufacturer/Type	Promise SX4000
# Participating Drives	4 ATA
Overview	During the process of constructing a RAID volume, not all of the available disk space is allocated or used. For example, in a RAID-5 using four 40 GB disks, there is approximately 64KB of disk space at the beginning of each drive that is not used by the actual RAID volume. Furthermore, the space is not simply ignored, but the RAID controller writes to it during the volume construction. This experiment was to understand what the controller actually does with the space, and to see if there is a discernable method and/or pattern to what is written.
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Promise SX4000 controller used. • 4 Barracuda 7200, 40 GB ATA(EIDE) drives (Channels 1, 2, 3, 4). • Each drive was wiped using DiskWipe (utility from NIST testing suite) and written with a specific value depending on the drive. o Drive #1, wiped with the value “A1”. o Drive #2, wiped with the value “B2”. o Drive #3, wiped with the value “C3”. o Drive #4, wiped with the value “D4”. • At the BIOS prompt, RAID-5 (Stripe/Parity) was setup utilizing all drives, yielding a 120 GB RAID-5 volume.
Created Partition Information	
Data Gathering	Each drive was examined using the DI-Tool-#2 boot CD. The drive was examined for changes to the original values written to each drive from the DiskWipe program (A1, B2, C3, D4).
Findings Comments	The RAID controller initialized the RAID-5 volume by modifying the first 65049 bytes of each participating drive in 4 distinct parts. (1) The first 30464 bytes (FO:00000 to FO:30464) were initialized to “00” by the RAID controller. (2) The next 16384 bytes (FO:30464 to FO:46847) was initialized with data, but it was unclear as to what specifically the data is, as its not a single value, and there is no discernable pattern. (3) The next 16384 bytes (FO:46848 to FO:63231) was again initialized to “00” by the RAID controller. However, within this section there were about 10 additional values randomly distributed (80h, 40h, 04h, 20h, 01h), some of which were repeated. (4) The last 1818 bytes (FO:63232 to FO:65049) was initialized with data, but again as in section 2, there is no discernable pattern to it. The remaining section of the drive had values written to it by the drive wipe program.

7.9 RAID-DI-SC-09: RAID-1 Information Written During 2-Drive Volume Construction

Test Case Identification	RAID-DI-SC-09
RAID Implementation	Type 1 / Type 5
RAID Type	Hardware
Manufacturer/Type	Promise SX4000
# Participating Drives	2 ATA
Overview	This experiment is examining if the changes made to the initial 64KB of each participating drive changes depending on the type of RAID implemented. In this case, a RAID-1 mirror is setup, and the drives are examined.
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Promise SX4000 controller used. • 2 Barracuda 7200, 40 GB ATA(EIDE) drives (Channels 1, 2). • Each drive was wiped using DiskWipe (utility from NIST testing suite) and written with a specific value depending on the drive. o Drive #1, wiped with the value "A1". o Drive #2, wiped with the value "B2". • At the BIOS prompt, RAID-1 (Mirror) was setup utilizing both drives, yielding a 40 GB volume.
Created Partition Information	
Data Gathering	• Each drive was examined using DI-Tool-#2 boot CD, for changes made by the RAID controller. Essentially the drive was examined for the original values written to each drive from the DiskWipe program (A1, B2).
Findings	The RAID controller initialized the RAID-1volume in the same format as the RAID-5 array. The first 65049 bytes on the drive were divided into 4 distinct parts. (1) The first 30464 bytes (FO:00000 to FO:30464) were initialized to "00" by the RAID controller. (2) The next 16384 bytes (FO:30464 to FO:46847) was initialized with data, but it was unclear as to what specifically the data is, as its not a single value, and there is no discernable pattern. (3) The next 16384 bytes (FO:46848 to FO:63231) was again initialized to "00" by the RAID controller. However, within this section there were about 10 additional values randomly distributed (80h, 40h, 04h, 20h, 01h), some of which were repeated. (4) The last 1818 bytes (FO:63232 to FO:65049) was initialized with data, but again as in section 2, there is no discernable pattern to it.
Comments	

7.10 RAID-DI-SC-10: RAID-5 Information Written During 3-Drive Volume Construction

Test Case Identification	RAID-DI-SC-10
RAID Implementation	Type 5
RAID Type	Hardware
Manufacturer/Type	Promise SX4000
# Participating Drives	3 ATA
Overview	This experiment was to see if the number of drives participating in a RAID-5 affects how the first 64k of the drives is initialized. In this case, three drives were used to construct the RAID-5, instead of four drives, and the changes made to the drives by the RAID controller are examined.
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Promise SX4000 controller used. • 3 Barracuda 7200, 40 GB ATA(EIDE) drives (Channels 1, 2, 3). • Each drive was wiped using DiskWipe (utility from NIST testing suite) and written with a specific value depending on the drive. o Drive #1, wiped with the value “A1”. o Drive #2, wiped with the value “B2”. o Drive #3, wiped with the value “C3”. • At the BIOS prompt, RAID-5 (Stripe/Parity) was setup utilizing all drives, yielding a 80 GB RAID-5 partition.
Created Partition Information	
Data Gathering	Each drive was examined using the DI-Tool-#2 boot CD. The drive was examined for changes to the original values written to each drive from the DiskWipe program (A1, B2, C3).
Findings	The RAID controller initialized the RAID-5 volume by modifying the first 65049 bytes of each participating drive in 4 distinct parts. (1) The first 30464 bytes (FO:00000 to FO:30464) were initialized to “00” by the RAID controller. (2) The next 16384 bytes (FO:30464 to FO:46847) was initialized with data, but it was unclear as to what specifically the data is, as its not a single value, and there is no discernable pattern. (3) The next 16384 bytes (FO:46848 to FO:63231) was again initialized to “00” by the RAID controller. However, within this section there were about 10 additional values randomly distributed (80h, 40h, 04h, 20h, 01h), some of which were repeated. (4) The last 1818 bytes (FO:63232 to FO:65049) was initialized with data, but again as in section 2, there is no discernable pattern to it. After these sections until the end of the drive, the remaining data is the pattern written to the drive from the DiskWipe program.
Comments	

7.11 RAID-DI-SC-11: RAID-5 Info Written During 4-Drive Volume Construction, 16k Striping

Test Case Identification	RAID-DI-SC-11
RAID Implementation	Type 5
RAID Type	Hardware
Manufacturer/Type	Promise SX4000
# Participating Drives	4 ATA
Overview	The standard configuration for RAID-5 is set for a 64KB stripe size. In this experiment, the stripe size is changed to 16KB to see if it has an impact on how all participating drives are initialized by the RAID controller.
Setup and Configuration	The setup of the target computer (Aqua) is as was defined in Figure 4. The RAID hardware and software configuration was as follows. • Promise SX4000 controller used. • 4 Barracuda 7200, 40 GB ATA(EIDE) drives (Channels 1, 2, 3, 4). • Each drive was wiped using DiskWipe (utility from NIST testing suite) and written with a specific value depending on the drive. o Drive #1, wiped with the value "A1". o Drive #2, wiped with the value "B2". o Drive #3, wiped with the value "C3". o Drive #4, wiped with the value "D4". • At the BIOS prompt, RAID-5 (Stripe/Parity) was setup utilizing all drives, yielding a 120 GB RAID-5 volume. The default stripe size was changed to 16KB for the RAID-5 volume.
Created Partition Information	
Data Gathering	Each drive was examined using the DI-Tool-#2 boot CD. The drive was examined for changes to the original values written to each drive from the DiskWipe program (A1, B2, C3, D4).
Findings	The size of the stripe (16KB) made no impact on how the RAID controller initialized the participating drives. The RAID controller initialized the first 65049 bytes of each participating drive in 4 distinct parts. (1) The first 30464 bytes (FO:00000 to FO:30464) were initialized to "00" by the RAID controller. (2) The next 16384 bytes (FO:30464 to FO:46847) was initialized with data, but it was unclear as to what specifically the data is, as its not a single value, and there is no discernable pattern. (3) The next 16384 bytes (FO:46848 to FO:63231) was again initialized to "00" by the RAID controller. However, within this section there were about 10 additional values randomly distributed (80h, 40h, 04h, 20h, 01h), some of which were repeated. (4) The last 1818 bytes (FO:63232 to FO:65049) was initialized with data, but again as in section 2, there is no discernable pattern to it. As before, after these sections, the drives all were unmodified and had their original wipe values to the end of the drive.
Comments	