Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

File recovery and data extraction using automated data recovery tools

A balanced approach using Windows and Linux when working with an unknown disk image and filesystem

Richard Carbone
Certified Hacking Forensic Investigator (EC-Council)
Certified Incident Handler (SANS)
DRDC Valcartier

Defence R&D Canada – Valcartier

Technical Memorandum
DRDC Valcartier TM 2009-161
January 2013

Canada

File recovery and data extraction using automated data recovery tools

A balanced approach using Windows and Linux when working with an unknown disk image and filesystem

Richard Carbone
Certified Hacking Forensic Investigator (EC-Council)
Certified Incident Handler (SANS)
DRDC Valcartier

Defence R&D Canada – Valcartier

Technical Memorandum
DRDC Valcartier TM 2009-161
January 2013

Principal Author

Original signed by Richard Carbone

Richard Carbone

Programmer/Analyst

Approved by

Original signed by Guy Turcotte

Guy Turcotte

Head/Mission Critical Cyber Security Section

Approved for release by

Original signed by Christian Carrier

Christian Carrier

Chief Scientist

Abstract

This memorandum is the direct result of the analysis of an unknown disk containing unknown data, files and filesystem. The disk was brought to an analysis team at DRDC Valcartier by an agency that desired to ascertain the research centre's capabilities for extracting and recovering unknown forensic data from an unknown disk and, if possible, automate the process. However, a thorough analysis using various Windows and Linux-based automated data and file recovery tools has led the author to determine that automated tools, regardless of the underlying system, are not yet up to this specific challenge. In addition, the author is of the opinion that fully automated disk recovery tools will never be entirely successful. Instead, the author has determined that a manual approach to data and file extraction will be necessary in order to recover any meaningful data or files from this disk's unknown filesystem. However, this memorandum will only examine the automated approach used by the various Windows and Linux tools. An additional follow-up study will specifically examine the required manual approach necessary for data recovery from an unknown disk using data pattern matching techniques and sector-by-sector analysis using known file signatures.

Résumé

Ce mémorandum est le résultat direct de l'analyse d'un disque inconnu contenant des données, des dossiers, et un système de fichiers obscur. Le disque a été apporté à une équipe d'analyse à DRDC Valcartier par une agence qui a désiré établir les capacités de centre de recherches pour extraire et en récupérant des données d'informatiques légales inconnues à partir d'un disque inconnu et, si possible, automatisez le processus. Cependant, une analyse complète utilisant divers outils de reconstitution de fichier et des données automatisées pour Windows et Linux a mené l'auteur à déterminer que les outils automatisés, indépendamment du système sous-jacent, ne sont pas encore jusqu'à ce défi spécifique. En outre, l'auteur est de l'opinion que les outils entièrement automatisés de récupération de disque ne seront jamais entièrement réussis. Au lieu de cela, l'auteur a déterminé qu'une approche manuelle aux données et à l'extraction de dossier sera nécessaire afin de récupérer tous les données ou dossiers indicatifs à partir de système de fichier obscur de ce disque. Cependant, ce mémorandum examinera seulement l'approche automatisée employée par les divers outils de Windows et de Linux. Une étude complémentaire examinera spécifiquement l'approche manuelle exigée nécessaire pour la récupération de données à partir d'un disque inconnu utilisant des techniques de configuration avec un modèle de données et l'analyse secteur par secteur utilisant les signatures connues de dossier.

This page intentionally left blank.

Executive summary

File recovery and data extraction using automated data recovery tools: A balanced approach using Windows and Linux when working with an unknown disk image and filesystem

Carbone, R.; DRDC Valcartier TM 2009-161; Defence R&D Canada – Valcartier; January 2013.

During the month of April 2009, DRDC Valcartier was approached by an agency to forensically analyse a disk containing unknown data, files and filesystem. The agency desired to ascertain whether DRDC could develop a simple automated approach for extracting data and files from analogous disks containing similar datasets. The agency is currently performing manual data extraction from suspect disks and would prefer spending more time analysing data rather than extracting it.

Upon completing a thorough analysis of Windows-based forensic data recovery tools, it was determined that none of them was capable of extracting any useful data from the unknown disk. Although some of the tools found more data than others did, they all came up short as none of the extracted data proved useful. At this time, various Linux-based tools were assessed for their ability to extract data from the unknown disk. Although very capable in their own right, the Linux tools also came up short, finding nothing of value. Thus, at this time, the author is unable to definitively conclude if an obscure filesystem containing unknown files with no known file signatures or structures is recoverable using an automated approach to data recovery. Instead, the author considers a manual approach to be more appropriate in such a scenario. However, depending on how a manual data recovery is carried out portions of it may be automatable.

The significance of this memorandum is to demonstrate that although automated data recovery tools are a valuable component in the tool chest of the forensic investigator, there are times when these tools, as good as they are, will not be capable of extracting any useful data or information. At this time, it is important that the forensic investigator be able to carry out a manual analysis of an unknown disk in order to attempt to extract meaningful data from it. However, the steps required to carry out a manual disk analysis are outside the scope of this specific memorandum. Instead, this memorandum will provide details on the various steps taken to find and extract data using automated recovery tools.

However, a future follow-up report will be written that directly examines the manner in which a manual data recovery can be carried out using data pattern-based analysis techniques. In the meantime, this memorandum provides ample information to the reader on how to successfully recover data files using specially designed automated tools.

Sommaire

File recovery and data extraction using automated data recovery tools: A balanced approach using Windows and Linux when working with an unknown disk image and filesystem

Carbone, R. ; DRDC Valcartier TM 2009-161 ; R & D pour la défense Canada – Valcartier; janvier 2013.

Pendant le mois d'avril 2009, DRDC Valcartier a été approché par une agence pour faire une analyse d'informatique légale à un disque contenant des données, des dossiers et un système de fichiers inconnus. L'agence a désiré s'assurer que DRDC pourrait développer une approche automatisée simple pour extraire des données et des dossiers à partir des disques analogues contenant les ensembles de données semblables. L'agence actuellement exécute l'extraction de données manuelle à partir des disques suspects et la préférerait passer plus de temps analysant des données plutôt que l'extrayant.

Lors d'accomplir une analyse complète des outils légaux basés sur Windows de récupération de données, on l'a déterminé qu'aucun de eux n'était capable d'extraire n'importe quelles données utiles à partir du disque obscur un système de fichiers inconnu. Bien que certains des outils aient trouvé plus de données que d'autres ont fait, elles toutes ont monté sous peu pendant qu'aucune des données extraites ne s'avérait utile. Actuellement, de divers outils basés sur Linux ont été évalués pour que leur capacité extraie des données à partir du disque inconnu. Bien que très capables de leur propre chef, les outils de Linux également ont monté sous peu, ne trouvant rien de valeur. Ainsi, actuellement, l'auteur ne peut pas conclure définitivement si un système de fichiers obscur contenant les dossiers inconnus sans les signatures ou les structures connues de dossier est récupérable utilisant une approche automatisée à la récupération de données. Au lieu de cela, l'auteur considère comme étant une approche manuelle plus appropriée dans un tel scénario. Cependant, selon comment une récupération manuelle de données est effectuée des parties de elle peut être automatable.

L'importance des ce mémorandum est pour démontrer que bien que les outils automatisés de récupération de données soient un composant précieux dans le coffre d'outil de l'investigateur légal, il y a des périodes où ces outils, aussi bons qu'ils sont, ne sera pas capable de n'extraire aucunes données ou information utiles. Actuellement, il est important que l'enquêteur d'informatiques légal puisse effectuer une analyse manuelle d'un disque inconnu afin d'essayer d'extraire des données indicatives à partir de lui. Cependant, les étapes exigées pour effectuer une analyse manuelle de disque sont hors de portée de ce mémorandum. Au lieu de cela, ce mémorandum fournira des détails sur les diverses mesures prises pour trouver et extraire des données utilisant les outils automatisés de récupération.

Cependant, on rédigera un futur rapport complémentaire qui examine directement la façon dont une récupération manuelle de données peut être effectuée utilisant des techniques basées sur modèle d'analyse de données. Dans le même temps, ce mémorandum fournit des informations suffisantes au lecteur sur la façon dont récupérer avec succès des fichiers de données utilisant les outils automatisés particulièrement conçus.

This page intentionally left blank.

Table of contents

List of figures

List of tables

Acknowledgements

The author would like to thank Mr. Yves van Chestein for reviewing this document and providing comments and corrections for it in its Technical Note incarnation.

The primary author would also like to extend a large thanks to Philippe Charland who helped provide the necessary comments and corrections in order to reincarnate this document as a Technical Memorandum. Moreover, the primary author would also to thank him for peer-reviewing this document

1 Background

1.1 Objective

The objective of this memorandum is to report on the findings that followed the analysis of an unknown disk provided to DRDC Valcartier by an external agency. The disk contains obscure files, data and filesystem, all of which have been assessed as unrecognizable by the author. More specifically, the agency desired to ascertain whether the research center would be able to devise a method for automating the extraction of data from an unknown given disk device instead of using a manual process. Manual data extraction is a highly time consuming and labour intensive process that takes vital time away from useful data analyses. However, a thorough analysis of the disk by the author using various data recovery-based software tools has shown that an automated approach to data and file extraction using these tools is not readily feasible due to the over-reliance of predefined file and data structures by these tools.

The analysis itself is broken into two parts. The first part uses various Windows-based data recovery-based software tools to attempt to find and extract data and files from an unknown disk. The second portion of the analysis concentrated on the use of Linux-based data recovery tools. Unfortunately, even though each of the tools analysed in this memorandum are highly capable in their own right, each was incapable of extracting any useful data or files from the disk.

The results of the various tools used herein have been documented so that they could be easily shared with others who may find themselves in similar circumstances. This memorandum, however, is not the end of the investigation, but only the first half. The second half of this study will examine in a separate memorandum how to successfully find and extract data from an unknown disk with data and unknown structures using detectable patterns.

1.2 Acquiring a disk image

1.2.1 Obtaining the suspect disk

The suspect disk was handed directly over to a DRDC scientist by a representative of an external agency. The data and files on the suspect disk have been doctored but are very lifelike. The preservation of the chain of custody and evidence is important and constitutes an integral part of any computer forensics investigation. Therefore, all actions and operations undertaken by the author, currently a certified forensic investigator[1], have been fully documented in accordance with best practice chain of custody procedures.

The disk was received mid-April 2009. This disk was then transferred over to the forensic investigator April 21, 2009. These actions have been fully documented and kept in secure storage with the original suspect disk. An image of the suspect disk was taken shortly thereafter on the desk of the forensics investigator.

[1] The author is certified by the EC-Council as a Certified Hacking Forensic Investigator.

1.2.2 Disk specifics

1.2.2.1 Suspect disk

The specifics of the suspect disk are as follows and the actual device itself can be seen in Figures 1, 2 and 3.

Manufacturer: Western Digital

Serial Number: WCANKM424372

Model Number: WD2500JB-00REA0

Date of Manufacture: 27 January 2008

Size: 250 GB

Interface: IDE

Logical Block Addressing (LBA): 488,397,168 512-byte sectors

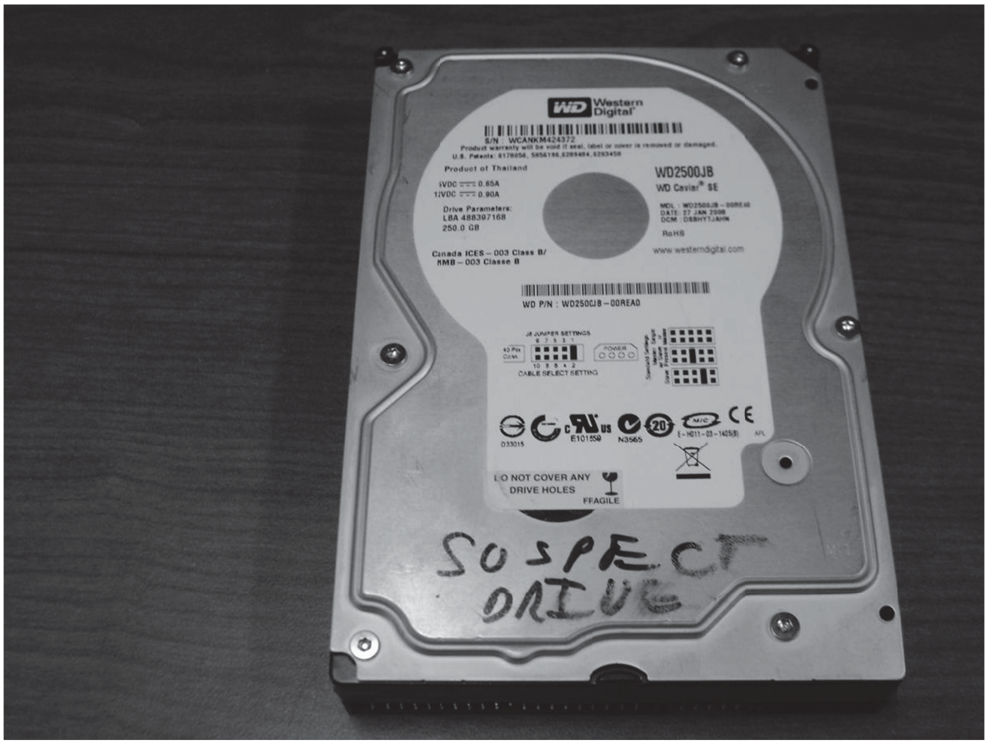

Figure 1: Top view of suspect disk

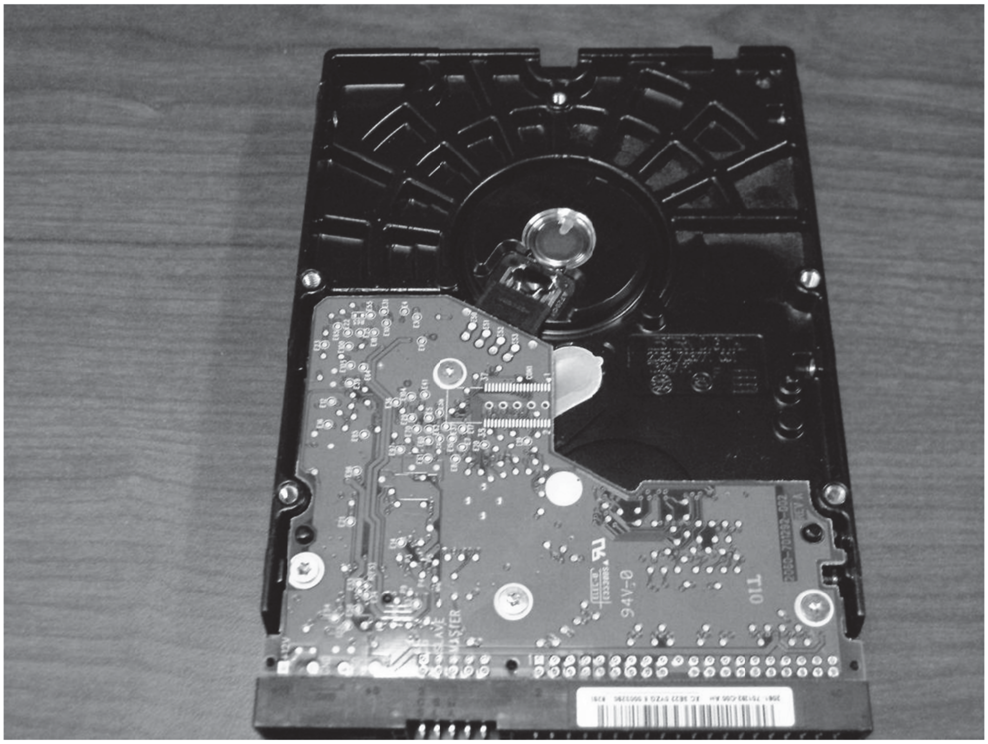

Figure 2: Bottom view of suspect disk

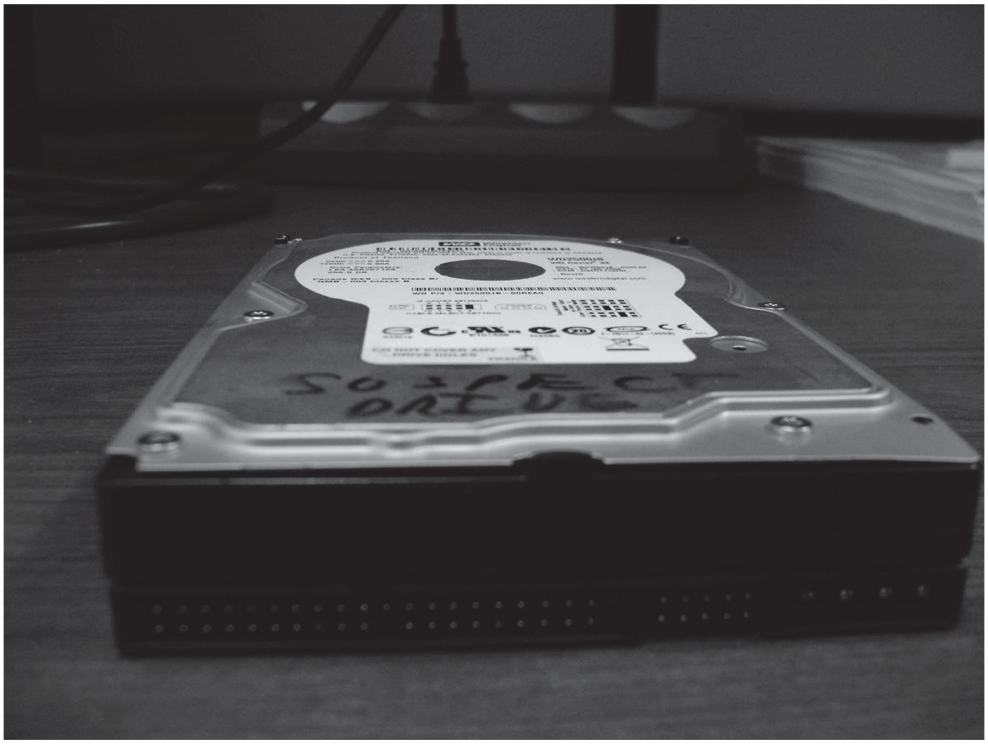

Figure 3: Side view of suspect disk showing disk interface

1.2.2.2 Target disk

The specifics of the target drive that will be used to store a disk image file of the suspect disk can be seen in Figures 4, 5 and 6. The target disk will store the entire contents of the suspect disk using a disk image file that is an exact bit-copy of the suspect disk's contents. The specifics of the target hard disk drive unit are:

Manufacturer: Hitachi

Serial Number: KRVN67ZBGHBGJF

Model Number: HDS725050KLA360

Date of Manufacture: February 2007

Size: 500 GB

Interface: SATA

LBA: 976,773,168 512-byte sectors

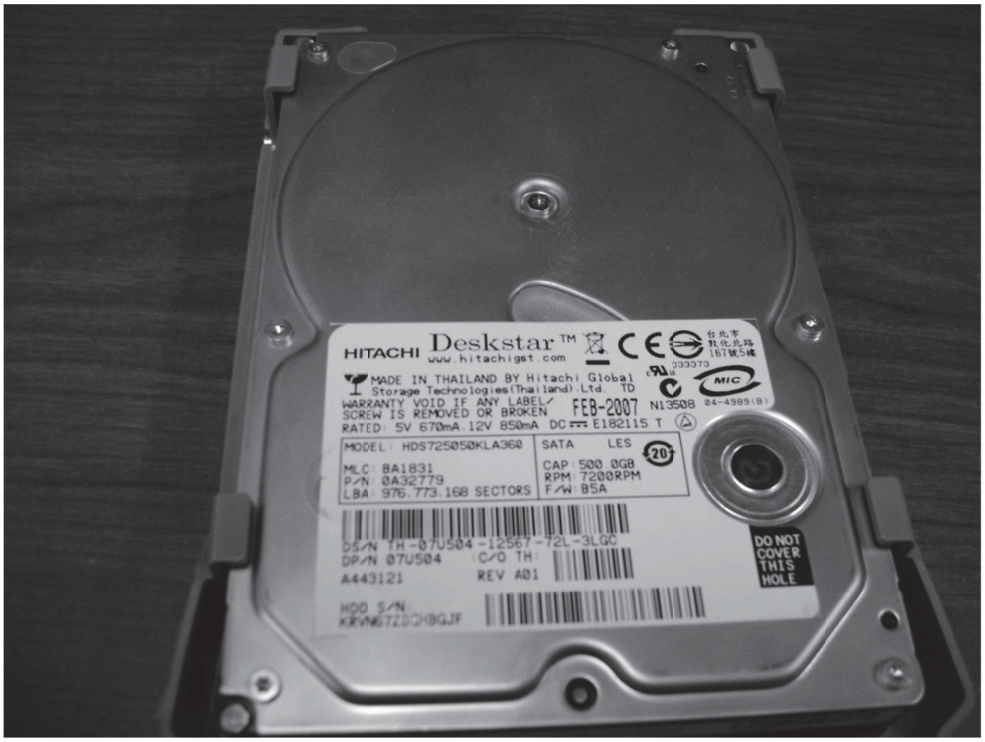

Figure 4: Top view of target disk

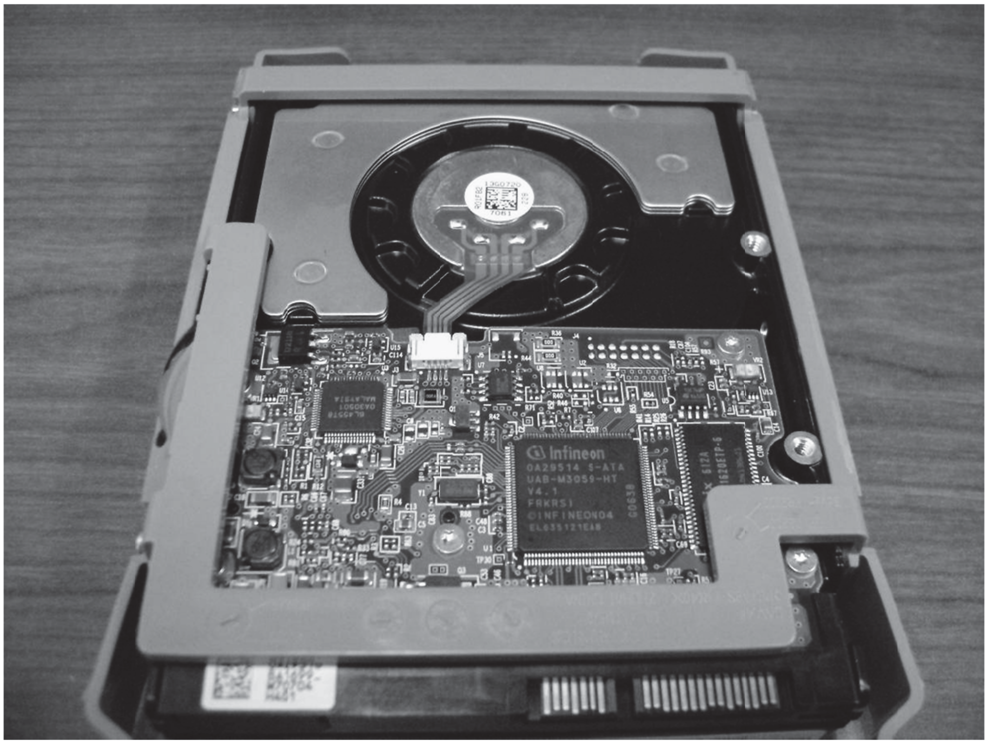

Figure 4: Bottom view of target disk

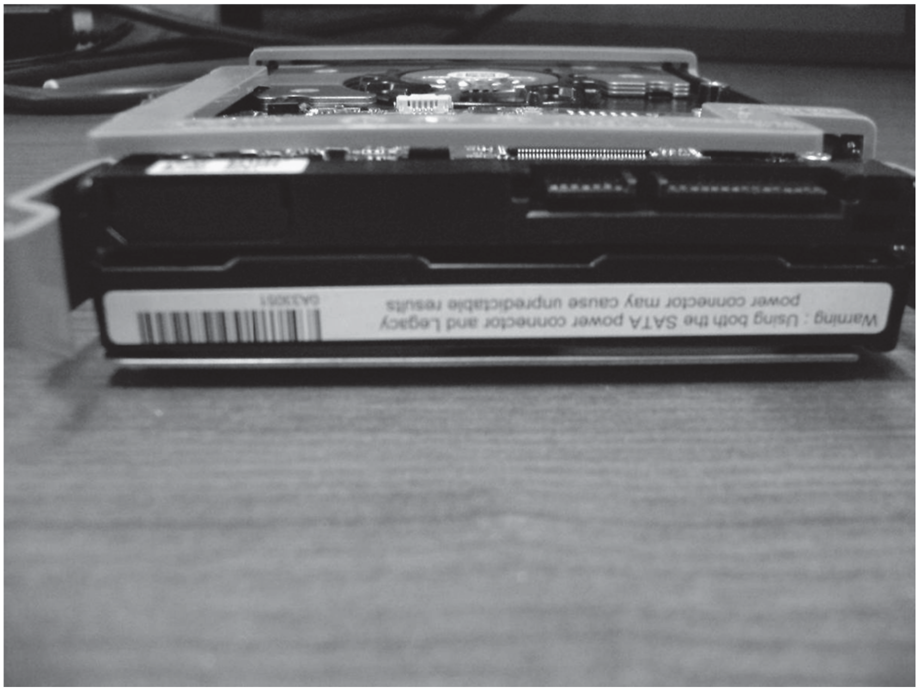
Warning : Using both the SATA power connector and Legacy power connector may cause unpredictable results

Figure 5: Side view of target disk showing disk interface

1.2.3 About the disk imaging computer

The forensics investigator currently has two very powerful computer workstations available for carrying out investigations. The first of these is a Linux-based computer system that is readily used as a disk imaging computer system. The specifics of the disk imaging computer system are provided below. Pictures of this system opened for immediate access to the machine's internals for inserting and removing disk drives can be seen in Figures 7 and 8. The disk imaging computer system is a Dell Precision 690 Workstation and consists of the following hardware:

2 dual-core (with HyperThreading) Xeon 3.20 GHz processors[2]

8 GB RAM

1 DVD-R/RW/RAM/CD-R/RW drive

1-500 GB Hitachi DeskStar system disk

1-3 ½" floppy disk drive

1 multi-port media card reader

2 FireWire ports

8 USB ports

1.2.4 Preparing the target drive

1.2.4.1 Connecting the target drive

The disk imaging computer system when used for Linux-based forensics investigations has a fully functional installation of 64-bit Fedora Core Linux 6 (FC6). However, the actual disk imaging process carried out herein was conducted using a bootable instance of Helix 3 Live CD on the disk imaging computer system.

Before any disk imaging could occur, it was necessary to remove the disk imaging computer system's FC6 disk drive and in its place insert the target drive that would store the disk image file of the suspect disk. The target drive was inserted into the top right-hand corner drive bay within the workstation (see Figure 7) where the FC6 originally resided.

[2] In effect, these HT-capable dual-core Xeon processors behave as if they were quad-cores and all multi-core operating systems including Windows XP and Linux detect 8 distinct processor cores. Therefore, for all useful purposes, the computer system is in effect taking advantage of 8 processing cores.

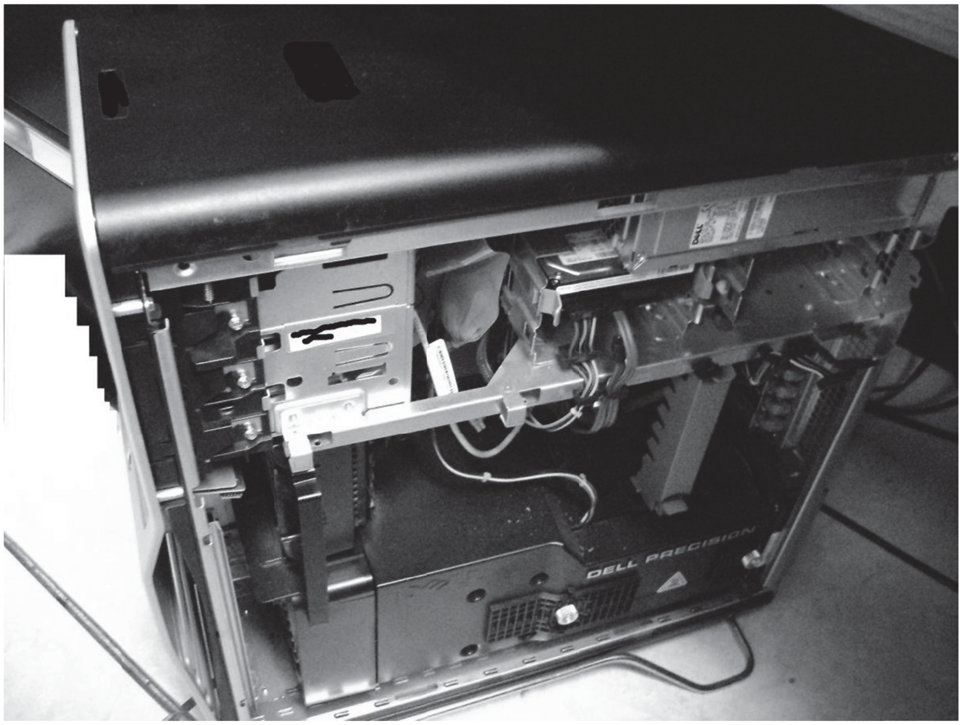

Figure 6: Disk imaging system's internals

Figure 7: Front view of the disk imaging system

The SATA and power cables were then connected to the target drive and the computer's chassis was closed (Figure 8). Upon securely closing the system chassis, the computer system was powered on. These actions were carried out at 11:25 am April 21, 2009.

1.2.4.2 Wiping the target drive

After removing the disk imaging computer system's FC6 operating system disk and inserting the target disk into it, the target drive had to be wiped and sanitized in order to remove any possible source of contamination from it. With the disk imaging computer system's chassis closed up, a copy of Helix 3 Live CD ready to be inserted into the CD drive, the computer system was powered up, and the CD inserted into the CD drive. The F12 key was pressed in order to induce the system to present a boot menu. Once presented with the boot menu, the computer was booted from the CD drive. After Helix 3 had completely loaded and X Windows was working correctly, a terminal window was opened and the following commands were executed to verify that the target disk was in fact present and then to wipe it:

```
(1) $ fdisk -lu
```

This command verified that the target drive had been successfully detected. It was detected as device */dev/sda*. The target disk was then wiped using a zero-fill wipe as follows:

```
(2) $ dd if=/dev/zero of=/dev/sda
```

A zero-fill wipe fills a target device with a pattern of binary zeroes, thereby wiping and sanitizing it. Since the data that previously resided on the disk was unclassified, a DoD-compliant wipe was not necessary. These actions were taken at 11:35 am April 21, 2009. The zero-fill of the target drive took slightly more than three hours to complete and no errors were detected or reported by the *dd* program.

1.2.4.3 Partitioning and formatting the target drive

Once the disk wipe of the target drive had completed, it was time to partition and format it. It was decided early on that a FAT32 partition and filesystem would be used since Linux easily supports it and Windows can only support NTFS and FAT-based partitions and filesystems. With the disk-imaging computer still powered on, the following actions and commands were taken in order to partition and format the target disk:

```
(3) $ fdisk /dev/sda
```

Within *fdisk*, a single FAT32 partition was created as */dev/sda1*. The partition */dev/sda1* was formatted as FAT32 using the following command:

```
(4) $ mkfs.vfat -F 32 /dev/sda1
```

After formatting the target drive to FAT32, the target drive was ready to accept data and files. These commands were executed in the early afternoon of April 21, 2009.

1.2.5 Preparing the suspect disk

1.2.5.1 Connecting the suspect disk

Since the target drive had already been inserted into the disk imaging computer system, as well as had been sanitized, partitioned and formatted, it was time to connect the suspect disk to the disk imaging computer system. However, the computer would have to be powered off and then on again in order to detect the suspect disk. The suspect disk's use of an IDE connection could have been problematic had a USB-to-IDE adapter not been found in storage while the operations listed in the previous sections were underway. Once the adapter had been located, the suspect disk was connected to it (see Figure 9) and was then ready to be connected to the disk imaging computer system. It was not deemed necessary to use a write-blocker since Helix 3 Live CD was in use and it is known for maintaining the forensic integrity of all attached disk devices until instructed to do otherwise by the operator.

The suspect disk's IDE connector was attached to the USB-to-IDE adapter and then the adapter-to-disk power cable and disk connector were affixed to the suspect disk. Then, one end of the adapter's power cable was plugged into the wall and the other end was affixed to the adapter itself. The computer system was correctly shut down and the adapter connected to the disk imaging computer using a USB cable (see Figure 10).

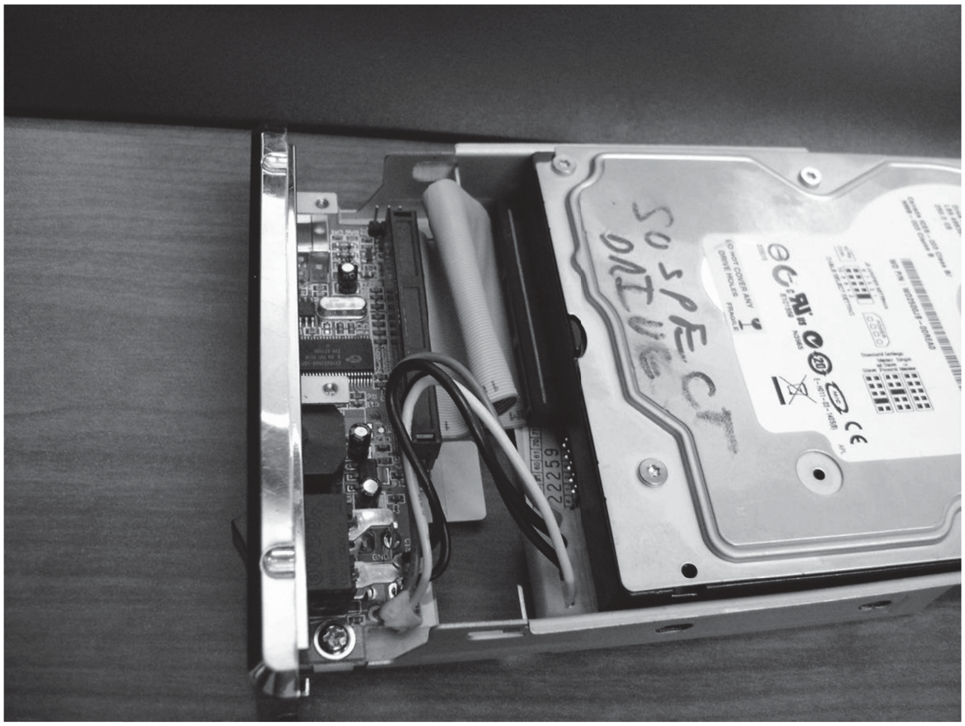

Figure 8: USB-to-IDE disk interface

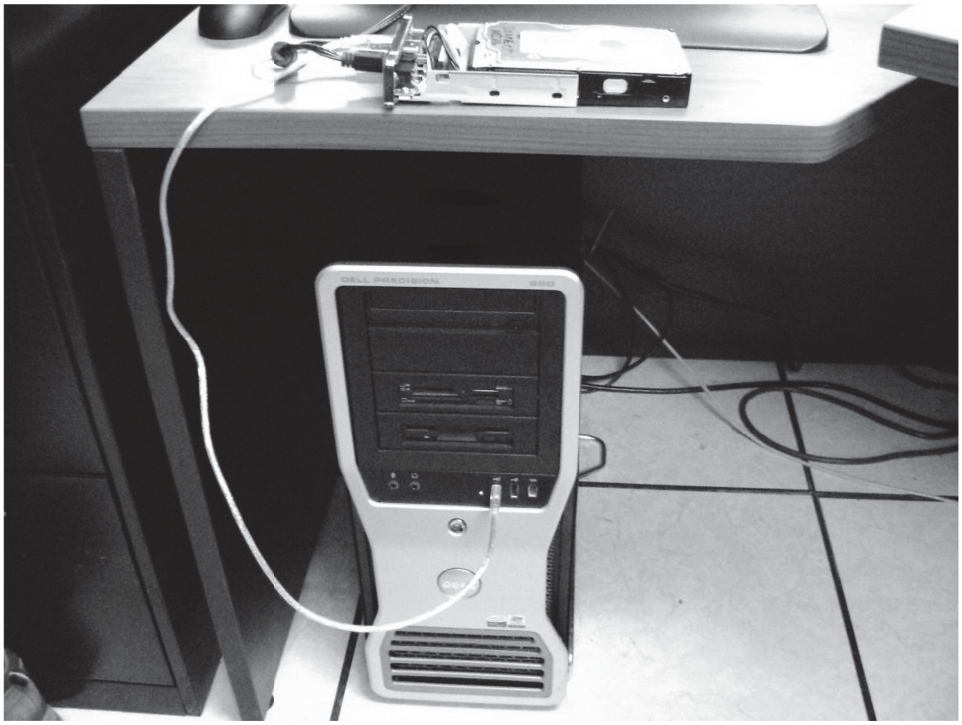

Figure 9: USB-to-IDE interface connected to the disk imaging computer system

The computer was then powered on again and the F12 key was pressed to obtain the system's boot menu. From there, the system was again booted from the Helix 3 Live CD, whereupon completion of its boot, X Windows was started. These events took place a few minutes after partitioning and formatting the aforementioned target disk during the afternoon of April 21, 2009.

1.2.5.2 MD5 hashing the suspect disk

With the suspect drive connected to the USB-to-IDE adapter and Helix 3 Live CD now fully booted, it was necessary to verify that both the target drive and the suspect disk had been detected. Using the following command from within a terminal window, both disks were identified:

```
(5)  $ fdisk -lu
```

Where the target drive was detected as */dev/sda* with a single FAT32 partition and the suspect drive had no recognizable partition on device */dev/sdb*.

At this time, an MD5 hash of the suspect disk *(/dev/sdb*) was to be generated. This was achieved using the following command:

```
(6)  $ cat /dev/sdb | md5sum > /tmp/suspect.md5
```

Where */tmp* was located as a part of the Live CD's filesystem. The */tmp* directory is commonly used in UNIX systems for temporary data storage. These commands were carried out at 3:01 pm April 21, 2009 and the MD5 hash completed several hours later that evening.

1.2.6 Disk imaging and verification

1.2.6.1 Imaging the suspect disk

The next day, it was seen that the MD5 hash of the suspect drive had successfully executed and completed sometime the previous evening and had been stored to the file */tmp/suspect.md5*. Because the target drive (*/dev/sda1*) had already been prepared for use the day before, it only had to be mounted before data could be written to it for storing the suspect disk (*/dev/sdb*) as a disk image file. Mounting the target drive read/write was done using the following commands:

```
(7)  $ mkdir /media/sdb1

(8)  $ mount /dev/sda1 /media/sda1
```

Carrying out a bit-copy of the suspect disk and saving it to the target disk was carried out using the following command:

```
(9)  $ dd  conv=noerrer  if=/dev/sdb  bs=512  of=/media/sda1/suspect.dd
```

Commands (7), (8) and (9) were the morning of April 22, 2009 and took the entire working day to complete. The disk transfer rate from the suspect disk to the target disk was significantly affected by the low-bandwidth inherent in both the USB-to-IDE adapter and the computer's USB port. No errors were detected or reported by the *dd* program.

1.2.6.2 MD5 hashing the disk image file

The next day, with the Helix 3 Live CD still running and the disk imaging having completed, an MD5 hash was made of the disk image file, */media/sda1/suspect.dd*, using the following commands that were run the morning of April 23, 2009:

```
(10)        $ cd /media/sda1

(11)        $ cat suspect.dd | md5sum > target.md5
```

Command (11) completed approximately 4.5 hours later after first beginning the MD5 hash of the acquired disk image file. The hash results were stored alongside the disk image in file */media/sda1/target.md5*.

1.2.6.3 Comparing MD5 hashes

In the early afternoon of the same day (April 23, 2009), upon completing MD5 hashing of acquired disk image, its hash was compared against the original MD5 hash (*/tmp/suspect.md5*) generated against the suspect two days prior. Both hashes were found to be the same using the following command:

```
(12)        $ diff /tmp/suspect.md5 /media/sda1/target.md5
```

The original hash of the suspect disk was copied to the target disk using the following command to preserve the evidence:

```
(13)        $ cp /tmp/suspect.md5 /media/sda1
```

In so doing, the target disk would then store the hash values for both the suspect disk and disk image file.

1.2.7 Completing the imaging process

Once the MD5 hashes had been compared and the integrity of the disk image file ascertained, it was time to shut down the disk imaging computer system. At 3:18 pm April 23, 2009, the disk imaging system was gracefully shutdown and the suspect disk given back to the scientist for secure storage. The target disk was also removed and locked up in the forensic investigator's secured filing cabinet. All notes taken were verified against the actions used to carry out the disk imaging process to determine if any faults or errors were present with the process used. It was determined that no faults or errors were present.

The disk imaging computer system's FC6 operating system disk was reinserted back into the machine for future use. The system was briefly powered up to ensure that it continued working. It was found to be in good working order, at which time the system was gracefully shutdown and left powered off.

1.3 Suspect disk origins and related technical details

According to the agent that brought the suspect disk to DRDC Valcartier, the disk originated from a security and surveillance DVR system. Specifics of the make and model will remain undisclosed so as not to jeopardize ongoing and future investigations. However, as with many other similar DVR models from competing manufacturers, they all implement specific proprietary technology. More specifically, this DVR implements unique and proprietary data compression, data encryption and watermarking technologies, thereby complicating data extraction and file recovery.

April 24, 2009, multiple in-depth Internet searches for the specific DVR model in question yielded little other valuable information. However, it was discovered that the filesystem used by the DVR device is entirely proprietary and that files could only be successfully extracted using the DVR remote user console interface necessary to use for viewing the various camera feeds and save them to a PC using commonly used video data formats. Alternatively, the DVR could also directly record video feeds to DVD disks inserted into the DVR storage device.

However, the objective of DRDC Valcartier was to determine how automated data extraction and file recovery could be carried out against such a disk and its data and files. Therefore, it was hoped that commonly used advanced data recovery tools, both Windows and Linux-based, would be able to extract some useful or meaningful data or files. This is described in the following two sections.

1.4 About system performance and disk analyses

The largest limiting factor for most data recovery operations using modern computers is not memory or processing-related. The bottleneck is generally caused by disk-based limitations. Most of today's commonly used hard disk drives have limited data transfer rates even though modern system buses can support larger transfer rates than what the disks can supply. In addition, modern system buses can support much larger transfer rates than that achieved by multiple disks in use simultaneously.

To better improve performance, it is worth considering the investment in high-end computer disk drives, especially when working with large amounts of data. At a minimum, 10,000 RPM SATA disk drives should be considered. Where even larger I/O requirements are necessary, multiple 15,000 RPM Ultra SCSI3 disk drives can be used in parallel, in conjunction with high-end SCSI host adapters. Other disk technologies include high-capacity RAID or fibre-channel disk devices connected to very-high speed PC host interface cards in order to dramatically boost disk-based performance.

Although computationally inferior, computer systems experience an improvement in performance when carrying out disk analyses upon upgrading memory and CPU. Even a modest disk upgrade will have a demonstrable effect on overall performance. It is important to understand that most disk analyses, even computationally intensive ones, tend to be more dependent on disk performance than they are on the processing capabilities of the system itself.

2 Windows tools for automated disk image analysis

Automated data recovery tools are best used when the underlying filesystem is either severely damaged or when working against an unknown or proprietary filesystem. For Windows-based data recovery platforms, many diverse data recovery-based software tools are available. In fact, there are more such tools Windows-based tools than for any other computer platform or architecture. Unfortunately, these tools are not all equal in capability and many of them implement similar functionalities thereby limiting the effectiveness of using multiple tools from disparate vendors. In addition, the origin of many of these tools is dubious at best. Furthermore, several of them offer little advanced data detection and recovery capabilities to the more knowledgeable user or forensic investigator.

The following Windows-based analysis represents a cross section of the various types of Windows data recovery tools currently available. All of these tools in some respect implement different functionalities and provide their inherent capabilities using differing mechanisms. Therefore, a thorough examination of an unknown disk using a representative cross section of modern data recovery Windows-based tools will enable the forensics investigator to ascertain the consistency of the recovered data. Moreover, this will further enable the investigator to determine whether data recovery is even possible, based on the results of the number and types of files recovered by the various tools undertaken in the endeavour.

2.1 About the Windows forensic workstation

The Windows forensics workstation used throughout this document consisted of a Dell Precision 690 workstation. It was configured as follows:

2 dual-core (with HyperThreading) Xeon 3.20 GHz processors

8 GB RAM

1 DVD-R/RW/RAM/CD-R/RW drive

1 CD/R/RW drive

1-500 GB Hitachi DeskStar system disk

1-3 ½" floppy disk drive

2 FireWire ports

8 USB ports

The installation of Windows on the aforementioned computer system was a fresh installation of Windows XP 64-bit Service Pack 2 and was patched with all current and relevant security updates and fixes. The system was installed with a full installation of Microsoft Office 2003 including the necessary Microsoft Office 2007 file format-based importation filters for Office 2003. A recent version of AVG Internet Security 8.5 was installed onto the workstation to scan and help prevent possible computer virus infection, either from the Internet or from any potential data recovered from the disk or disk image. In addition, the Windows multi-algorithm data-hashing tool MD5Deep was downloaded and installed onto the computer system.

In addition, software including WinRAR, Norton Ghost 9.0, OpenOffice 3.0.1, the Gimp 2.6.6, and XnView 1.96.1 were downloaded from the Internet and then installed onto the computer system. The various Windows data recovery software used herein were all downloaded from the Internet and installed onto the workstation.

Once the data recovery was underway using the various data recovery tools (see Section 2.3), the computer system was altogether disconnected from the network throughout the duration of the analysis (Section 2) and the Linux-based analysis described in Section 3. If any additional software had to be installed, it was done from scanned original media from the vendor.

A bootable Live Windows CD equipped with a recent version of AVG Antivirus and a very recent antivirus database was booted in order to scan the forensics workstation to ascertain if any malware had contaminated the system. Fortunately, none was found. At that time, the computer was powered off, pending the rest of the process.

These actions took the entire working day of April 24, 2009 to put into place.

2.2 Creating a secondary suspect disk device

Not all of the Windows-based data recovery tools were capable of working with disk image files. Capable tools included R-Studio, Zero Assumption Recovery and DiskInternals Partition Recovery. The other tools required direct disk access in order to perform their analyses. Direct disk access is achieved by working directly with physical raw disk devices. This section details how a raw second suspect disk copy was created. The preservation of evidence is paramount. Working with the original suspect other than for generating the first preliminary copy goes against this most important of computer forensics rules.

Therefore, a second suspect disk had to be created. It could have been created directly from the original suspect disk or from the validated disk image file already stored on the primary target drive (Section 1.2.7). The author decided that since a validated disk image file already existed, it made sense to generate a secondary raw disk device based on this disk image file. Preferably, an exact copy (at least in terms of drive geometry) of the suspect disk would be used as the secondary target drive. Fortunately, a SATA-enabled Western Digital hard disk drive with the same disk geometry and size was found in storage and was used for transferring the disk image. This SATA-enabled drive was from Western Digital's WD2500AAKS family line-up of 250 GB SATA drives.

Using the disk imaging computer system in conjunction with the Western Digital 250 GB SATA secondary target disk and the Hitachi DeskStar primary target disk device that stored the disk image, transferring the image to secondary disk device would be straightforward to carry out. The disk imaging-based computer system, still powered off from the last time it was used, had its chassis opened and the Hitachi DeskStar primary target drive (see Section 1.2.6) as inserted into the machine's secondary hard disk drive bay. The Western Digital 250 GB SATA secondary target drive was inserted into the machine's third hard disk drive bay. The appropriate power and disk cables were attached to the two inserted disk drives, the chassis was closed and the computer was powered on. The computer's FC6 operating system disk, sitting in the first disk drive bay, was booted up into runlevel 1 where the command string "LINUX S" was appended to the GRUB boot loader's kernel boot-up string.

Once the system's console was available, using the "fdisk -lu" command, the status of three disk drives was verified. The FC6 operating system disk was detected as */dev/sda*, the primary target drive as */dev/sdb* and the secondary target drive (the Western Digital SATA 250 GB drive) as */dev/sdc*. The following commands were used to mount the primary target drive, and then transfer the disk image file and checksum validate the transferred data:

```
(14)        $ mkdir /media/sdb1

(15)        $ mount /dev/sdb1 -ro /media/sdb1

(16)        $ dd  conv=noerrer  if=/media/sdb1/suspect.dd  bs=512  of=/dev/sdc

(17)        $ cat suspect.dd | md5sum > /tmp/secondary.md5

(18)        $ diff /media/sdb1/target.md5 /tmp/secondary.md5
```

Running command (18), no differences were discernible between both MD5 hashes, where the file *target.md5* denotes the already verified MD5 generated from transferring the raw suspect disk device to the disk image file stored onto the primary target disk device. The file *secondary.md5* denotes the MD5 hash generated for the disk image file data transferred directly to the raw secondary target drive-based storage device. There was no need to wipe the secondary target disk drive, as the data from disk image file would write directly over any pre-existing data, if any. Nevertheless, the generation of the MD5 hash validates that the disk image file was correctly transferred to secondary target disk device. The MD5 hash file *secondary.md5* was then transferred to the primary target drive for permanent storage using the following commands:

```
(19)        mount -o remount,rw /dev/sdb1 /media/sdb1

(20)        mv /tmp/secondary.md5 /media/sdb1/

(21)        umount /media/sdb1
```

These operations took almost an entire day to complete. Work began the morning of April 27, 2009 and finished towards the end of the workday. Once all the operations had successfully completed, the disk imaging computer system was powered off and the two target drives removed from the computer system. At this time, they were introduced into the Windows forensics workstation that had been left powered off until that time. With the primary and secondary drives inserted into the Windows forensics workstation's (occupying the system’s second and third disk drive bays, respectively) the system was turned. The primary target drive was detected as drive F:\, while the secondary target drive was detected but not allocated a drive letter, as it did not have a Windows recognizable filesystem.

2.3 The tools used and other conducted procedures

2.3.1 R-Studio

R-Studio, made here in Canada, is highly advanced data recovery software that supports a very large cross section of modern operating systems. It provides full data recovery support for all implementations of FAT and NTFS, as well as full support for Ext2/3, HFS/HFS+ and UFS1/UFS2 filesystems, enabling it to recover deleted data from a variety of operating systems and filesystems. In fact, R-Studio is, in the opinion of the author, the most versatile Windows-based data recovery tool examined throughout this memorandum. The tool supports data recovery using either disk image files or from raw physical devices and provides recovery support for more than 300 specific file types, more than any other data recovery tool, Linux or Windows-based, except for Ontrack EasyRecovery Professional, which supports just over 400 file types. The tool works well against damaged and corrupted filesystems, even if a filesystem has been replaced with an altogether different one (e.g. NTFS replaced by Ext3 filesystem). It also supports the ability to recreate RAID volumes based on disk image files.

R-Studio version 4.6 was used against the disk image file found on the primary target (*F:\suspect.dd*). It took approximately 3.5 hours to fully scan the disk image file. It detected 210 files. Specifically, it found 63 HA (Hyper Archiver) archives, 54 Pack archives, 6 Win-Freeze archives, 30 Windows Clipboard files, 55 SCO archives, and 2 PPM graphics images. The files were all extracted to *C:\TEMP* from where they were then analysed.

As determined from web searches using a different network-enabled computer system, many of these file types could not be opened or tested as their creating programs are no longer readily available, including HA archive files, ICE archives (similar to LHA), Pack archives and SCO archives. Other files, Windows Clipboard files and PPM image files were found to be corrupt using the Windows XP built-in clipboard viewer and the Gimp version 2.6.5, respectively.

R-Studio did not detect any filesystem structure, only file-like signatures, which it mistakenly interpreted as valid file structures. However, all post-extraction analysis determined that none of the files recovered were in the least but related to the DVR. Therefore, based solely on the results of this program, it could be concluded that no useful or discernible data could be extracted from the disk image using this program.

The program was started at the beginning of the workday of April 28, 2009 and completed for lunchtime of that same day. Analysis of the extracted files was completed before the end of the workday.

2.3.2 Zero Assumption Recovery

Zero Assumption Recovery is a powerful and popular data recovery application. It supports the analysis of disk image files and raw disk devices. Version 8.4 Build 15 of the software was used for this analysis. Based on information obtained from the company, it fully supports all FAT and NTFS partitions and filesystems, including the recovery of deleted or damaged partitions. However, it does not directly support other filesystems, although its file carving capabilities do provide the ability to recover data regardless of the underlying filesystem. It also supports a large collection of file signatures, although the exact amount was not available, even after requests to the company. In addition, the tool also supports the reconstruction of RAID volumes from disk image files. The tool was run directly against the disk image file *F:\suspect.dd*.

After starting the tool, the option "Recover data from a simple volume or a functional RAID volume" had to be selected in order to configure the program to load a disk image file. After accepting the selection, the image was loaded and the scan began. Approximately 2 hours later, the entire scan had completed but no file or data of any sort had been detected or recovered. The program was launched at the end of lunchtime April 28, 2009 and finished in the mid-afternoon of that very same day.

2.3.3 Stellar Phoenix Photo Recovery

Stellar Phoenix Photo Recovery is a popular photo and image recovery tool capable of recognizing and restoring diverse image formats. The tool supports 42 specific image file formats. However, the tool will not work on non-Stellar Phoenix (company specific proprietary format) compatible disk image files, although both Stellar Phoenix Photo Recovery and Stellar Phoenix Windows Data Recovery are capable of generating these disk image files from a raw disk device. Therefore, because the disk image file was created using *dd*, the secondary target disk drive generated in Section 2.2 was needed to perform an analysis with this tool. Stellar Phoenix Photo Recovery was then run directly against the secondary raw disk device.

Stellar Phoenix Photo Recovery version 3.1.0.0 was used for this analysis. It supports the recovery of many popular and common digital camera image formats, including TIFF, JPEG and camera manufacturer specific formats, as well as various formats multimedia formats including MP3, AIFF, MPEG and many others. Camera manufacturer support is provided for Nikon, Canon, Minolta, Kodak and several other manufacturers' proprietary formats.

However, after running the tool, not a single image or multimedia file of any sort was recovered. At the time, using this data recovery tool appeared as a potentially successful avenue of approach since the DVR device from which the suspect disk originated records multimedia-based data. However, such was not the case. Therefore, based on the results obtained using this tool, it could be concluded that the DVR device in question does not record data using any of the commonly supported multimedia formats supported by this tool. As the DRV device records data using an encrypted, proprietary, watermarked format, it should come as no surprise that the tool did not succeed. The process was initiated at the beginning of the workday April 29, 2009 and completed towards the end of the morning of that same day.

2.3.4 Stellar Phoenix Windows Data Recovery

The tool supports more than 250 different file formats. However, as with Stellar Phoenix Photo Recovery, Stellar Phoenix Windows Data Recovery does not offer the ability to work with non-Stellar Phoenix disk images including *dd*-based disk images. Therefore, because the disk image file was created using *dd*, the secondary target disk drive generated in Section 2.2 was needed to perform an analysis with this tool. Stellar Phoenix Windows Data Recovery was then run directly against the secondary raw disk device.

Unlike the other tools to date, Stellar Phoenix Windows Data Recovery version 3.0.0.2 was able to find and recover 344 files occupying 445.06 MB of disk space from the secondary target disk. The recovered data was saved to *C:\TEMP*. However, after analysing all the files that were found and recovered, it was determined that none of them were valid. They were all corrupt and unusable.

Of the various files found, multiple Zip and ARJ-compressed archives were extracted from the raw disk device. These files were tested using WinRAR version 3.80. In addition, one instance of a BZ2-compressed archive was extracted and tested using WinRAR. The recovery tool also found various graphic images including CAD 3D, PPM, PGM, PIC and PAL (Dr. Halo) file formats, all of which were analysed using XnView version 1.96.1. The tool also detected several other file formats including EFX (e-Fax), FLC (Autodesk animations), and HA (Hyper Archiver) archives. As determined from web searches using a different computer system, only the HA archives could not be analyzed since its creating software could no longer be found. The other two files, the EFX and FLC files were analysed using XnView. All successfully analysed files (with the exception of the three HA archives) were found to be corrupt.

The tool was initiated just moments after the previous tools and completed towards the end of the afternoon of tat very same day (April 29, 2009). The analysis of the files recovered took place throughout the following workday.

2.3.5 DiskInternals Partition Recovery

The first action undertaken using the DiskInternals Partition Recovery tool was to search for and if possible, recover and restore any lost or damaged partitions. DiskInternals Partition Recovery version 2.92 supports two main features. The first is partition recovery and the second is data recovery. The exact number of supported file types is unfortunately unknown for this software. The tool can be used against both disk image files and raw devices. For this specific software analysis, the disk image file *F:\suspect.dd* was selected.

Unfortunately, the partition recovery portion of the analysis yielded no useful results. In fact, it found no supported partition type at all on the disk image file, even though the software fully supported all FAT, NTFS and Ext2/3 partitions. The partition recovery analysis required approximately two hours.

The next portion of the analysis required that the forensic investigator select the type of filesystem to restore. This is odd since none of the other tools requested the operator to specify the underlying type of filesystem. However, the investigator chose to implement one recovery for each supported filesystem type. This yielded three analyses in all, one for NTFS, one for FAT and the final one for an Ext2/3 filesystem recovery. Each different analysis always yielded the same results: five Windows PE executables and seven JPEG images. Each recovered file was corrupt and unusable. The PE executables were tested from the Windows XP command line interface (CMD.EXE) and the JPEG images were examined using XnView version 1.96.1. In addition, each analysis (NTFS, FAT and Ext2/3) took an additional two hours.

The analysis was started at 7:45 am and was completed the same day at 5:30 pm. Analysis of the extracted files was carried out the morning of May 4, 2009.

2.3.6 Ontrack EasyRecovery Professional

Ontrack EasyRecovery Professional is highly reputable data recovery software developed by Ontrack, a pioneer in the electronic data recovery industry. The software is highly capable and supports the largest selection of file types available in any software examined throughout this memorandum.

The software, version 6.12, was the morning of May 4, 2009 and completed in the early afternoon of the same day. Ontrack EasyRecovery Professional software found the following file types:

Table 1. Files types recovered using Ontrack EasyRecovery Professional

File type found	No. of files found	Analysis program
dBase (*.DBF)	17,569	Analysed using OpenOffice version 3.0.1 based on a random selection of 400 files
Windows Clipboard (*.CLP)	30	Analysed using the Windows XP built-in clipboard viewer
Corel Paradox database (*.DB)	1	No viewer available for analysis
Micrografx (*.GRF)	17	No viewer available for analysis
Windows Icon (*.ICO)	20	Analysed using XnView version 1.96.1
GZIP (*.GZIP)	3	Analysed using WinRAR version 3.80
Z Compression (*.Z):	57	Analysed using WinRAR version 3.80
Autodesk AutoCAD (*.DWG)	10	Analysed using XnView version 1.96.1
Microsoft Write (*.WRI)	47	Analysed using Windows XP Write
SafeBack Disk Image (*.IMG)	30	Analysed using Linux Foremost
ARJ (*.ARJ)	25	Analysed using WinRAR version 3.80

Ghost (*.GHO)	50	Analysed using Norton Ghost 9.0
Microsoft Office File (*.MOF)	14	Analysed using MS Office 2007 to view files
Autodesk Animation (*.FLI)	47	Analysed using XnView version 1.96.1
Autodesk Animation (*.FLC)	21	Analysed using XnView version 1.96.1
Executable (*.EXE)	5	Ran using Windows XP command line interpreter (CMD.EXE)
Encapsulated PostScript (*.EPS)	10	Analysed using XnView version 1.96.1
Bitmap (*.BMP)	57	Analysed using XnView version 1.96.1

These files were extracted and saved to *C:\TEMP*. Although the data extraction was completed the same day as the analysis, the verification of the extracted files occurred over the next two days, May 5 and 6, 2009 in order to correctly ascertain the validity of these files. However, not all of the files could be immediately evaluated due to the lack of available software to analyse specific file formats, including Micrografx GRF image files and Paradox DB database files. All the remaining file formats were analysed and were determined to be corrupt or invalid as no useful information, meaningful data or images of any sort could be determined from the extracted files.

The SafeBack images that were recovered by the Ontrack software were analysed using the Linux Foremost program on the disk imaging computer system. However, the SafeBack data files had to be transferred to the disk imaging computer system using a DVD, since network connectivity could not be enabled so long as the analysis was not completed. The Ghost .GHO disk images were analysed using an already installed commercial copy of Norton Ghost 9.0.

Only two file formats, the first for Micrografx and the other for Corel Paradox, could not be viewed or analysed. However, based on the results from all of the other analysed files, it can be inferred that the two remaining unanalyzable file formats were invalid, as were all the successfully analysed file formats.

It is likely that the diverse array of files detected using Ontrack EasyRecovery Professional was made possible by the diverse file detection signatures the software supports, although in the end none of the files was valid. The same generalized conclusion can also be applied to all the dBase files, even though only a random sample consisting of approximately 2.3% (400 files) of the dBase file population was analyzed. A random sampling this size, although small, is accurate enough to be reasonably representative of the population as a whole.

2.4 Additional notes

Although not previously stated, all the Windows tools used herein, with the exception of DiskInternals Partition Recovery and Zero Assumption Recovery software, enable the user to explicitly configure the desired file types to recover. However, there is no uniform method for configuring the specific file types to recover using any of the aforementioned Windows-based tools. The following is a list of the Windows recovery software used herein enumerating the number for supported file types for each respective software tool:

R-Studio: 300+ file types

Zero Assumption Recovery: unknown

Stellar Phoenix Photo Recovery: 42 image file types

Stellar Phoenix Windows Data Recovery: 250+ file types

DiskInternals Partition Recovery: unknown

Ontrack EasyRecovery Professional: 400+ file types

Whenever possible, each tool was explicitly set by the user to search for and recover every possible file type that it could support. Some of the tools (e.g. R-Studio) by default had most but not all the file types selected for recovery. Others, including Stellar Phoenix's Photo Recovery and Windows Data Recovery, as well as Ontrack EasyRecovery Professional, were configured by default to recover all supported file types.

Statistically, random sampling can very accurately represent the overall population given that the size ratio between the sample and the overall population is significant enough to be meaningful. In truly random samples where bias is not a factor, a small sample of 1% to 2% the size of the population is meaningfully significant [1]. Randomly determining which specific files to analyze within a given file set was determined using a spreadsheet and various random number generation functions. Such a tool was used for analysing the data found in Section 2.3.6.

While it is a common practice in computer forensics to hash all extracted data and files from imaged disk image files and raw devices, this practice was not implemented herein. It was not deemed necessary, since none of the data recovered would be used beyond the extent necessary for research purposes.

2.5 Conclusion

Despite the capabilities of the various Windows data recovery tools examined herein, none was actually able to extract any useful files or data from either the suspect disk-based image file or the secondary raw disk device. Although four out of the six programs succeeded in detecting and extracting data either from the disk image file or from the secondary raw disk device, none of the tools was able to find and recover files or data of any value or use. All successfully analysed files turned out to be corrupt and of no particular use. Certain files types could not be analysed due to the lack of appropriate software for reading and examining them at the time of testing. However, based on the results from the successfully analysed files, it is logical to conclude that those unanalyzable files were also corrupt and therefore contained no useful information or data.

It was hoped at the outset of this analysis that using various Windows tools from differing vendors would have concluded with meaningful results. However, with no valid filesystem structure and no discernible files, the results are not very surprising. Based on obtained results and the frequency of repetition between certain file types regardless of the tool used, it can be posited that these tools, regardless of the underlying capabilities, are at their heart fancy file carving tools and that their underlying data recovery and restoration mechanisms are not all different from one another.

Therefore, based solely on the current results, it is uncertain if automated Linux-based data recovery tools will fare better.

3 Linux disk image analysis tools

3.1 About the Linux forensic workstation

3.1.1 Creating and configuring the Linux forensic workstation

The Linux forensic workstation used in this section is in fact a virtual workstation generated using Ubuntu Linux 8.10 in conjunction with VMware Workstation 6.5.2. This required installing a commercially licensed copy of VMware workstation 6.5.2 obtained from CD-based media and was virus and malware-scanned prior to its installation. The Windows forensics workstation remained disconnected throughout this section's analysis of the disk image file in order to prevent any potential external contamination of the system. The Linux Ubuntu virtual machine was created on an external FireWire 800 drive that was connected to the Windows forensics host (see Section 2.1) system using the appropriate cable connection.

While configuring the Linux virtual machine, a total of 2.5 GB of physical memory (RAM) was allocated to the virtual machine, as were two processing cores. The initial virtual machine disk file (VMDK) created was set to 70.0 GB in size. This virtual disk was partitioned and formatted without the use of any swap space and set aside as a single Ext3 partition and filesystem (*/dev/sda1*). Ubuntu was then installed onto this VMDK and was updated to a recent version from an antivirus scanned DVD of pre-downloaded updates made available from an altogether different computer system. The updates from the DVD were applied to the Ubuntu Linux installation. This entire process required approximately 4.5 hours. After the update, the Ubuntu Linux system was running kernel 2.6.27-11 i686 SMP.

Additional forensic software had to be installed onto the Ubuntu Linux installation including Testdisk and Photorec (both are part of the same software suite), Foremost, Recoverjpeg, Scalpel, Magicrescue and all of their required dependencies. These software packages were downloaded from a network-enabled computer and burned to a CD that was then antivirus scanned prior to loading it onto Ubuntu Linux. Downloading the various software packages and required dependencies did not take long, perhaps one hour to download and less than 30 minutes to install.

After the creation and configuration of the Ubuntu Linux installation, it was powered down so that a new 500 GB VMDK volume could be created in addition to the existing 70.0 GB VMDK. This operation took about 3 hours to complete. Once completed, the Ubuntu Linux virtual machine was powered on and several minutes later under X Windows, the new disk volume was configured. Under an X Windows terminal window, the new 500 GB disk volume was partitioned using the *fdisk* program resulting in a single Ext3 partition and filesystem, */dev/sdb1*. After formatting, the system was configured (by modifying */etc/fstab*) to automatically mount */media/sdb1*. A copy of the disk image file from the Windows forensics workstation, file *F:\suspect.dd*, was transferred to the virtual machine using FTP. The disk image file was saved to location */media/sdb1* as file *suspect.dd*. FTP was required for file transfers since VMware Tools had not been installed onto the virtual machine. Reasonable transfer rates of approximately 21.8 MB/s were achieved, requiring approximately 3.15 hours for transferring the 250 GB file from the host system to the virtual machine. The two MD5 hash files *suspect.md5* and *target.md5* were also uploaded to the virtual machine's 500 GB VMDK. However, the secondary raw target disk's hash file (*F:\secondary.md5*) was not transferred, as it was not needed.

Work on the Ubuntu virtual machine started early the morning of May 7, 2009 and had completed by the end of the day.

3.1.2 Verifying the integrity of the copied disk image file

At this time, it was approximately 4:30 pm May 7, 2009. Using standard Linux tools, the copied disk image file *F:\suspect.dd* has been transferred */media/sdb1* and saved as file *suspect.dd* using FTP. However, the transferred file had to be verified in order to ascertain its integrity and authenticity. This was done using the *md5sum* program against the file */media/sdb1/suspect.dd* and verifying its hash to the original suspect disk's hash stored in file */media/sdb1/suspect.md5*. The output from the *md5sum* program was saved to */media/sdb1/ftp.md5* and its contents compared to *suspect.md5* using the *diff* command. The following commands were used to carry out these actions:

```
(22)        $ cat /media/sdb1/suspect.dd | md5sum > /media/sdb1/ftp.md5

(23)        $ diff /media/sdb1/ftp.md5 /media/sdb1/suspect.md5
```

Command (22) was left to run overnight and upon returning to the office the next morning, the command has completed. The ensuing command (23) was run the next morning at 8:15 am May 8, 2009, and both hash files were found to contain exactly the same values, thereby authenticating the integrity of the transferred disk image file */media/sdb1/suspect.dd*.

3.1.3 Comments

All of the Linux tools run in this section were run concurrently from their own terminal window, as each program was able to work directly against the transferred disk image file. Furthermore, the advanced I/O capabilities of Linux were apt to handle the heavy data processing load placed on it by multiple concurrent disk I/O requests stemming from the various simultaneously running data recovery programs, thereby ensuring a file-lock and contention free analysis work environment.

The overwhelming majority of Linux-based (and UNIX) forensic and data recovery tools are capable of working with disk image files and the Linux tools used in this section are no exception. Of course, they will almost all work with raw disk devices too.

3.2 The tools used and other conducted procedures

3.2.1 Testdisk

Testdisk is an advanced UNIX-based partition recovery-based software tool that is capable of detecting and repairing more than 30 different partition types, far more than any known Windows software tool. It is also a part of the same software suite as Photorec. The software works equally well against both raw disk devices and disk image files. At the time of this writing, the most recent version of Testdisk, version 6.11.3, had been installed to the system under */opt/testdisk*. The software did not require compilation as the specific version downloaded had already been statically compiled for Linux 2.6.x kernels. Testdisk and Photorec are relatively simple tools to use and provide simple yet intuitive text-based interfaces that are easily navigated.

The Testdisk tool permits the user to select the type of system from where the raw disk or disk image file originated based on a menu of seven available choices. The fourth option, "None", was chosen, thereby indicating that the media either is not partitioned or damaged, corrupt or altogether proprietary in nature.

The software was run against the disk image file */media/sdb1/suspect.dd*. Upon starting the program, the "Proceed" option was selected, followed by "None", and then followed by "Options" to set the necessary search options. The tool's available options were set to:

Expert mode = Yes

Cylinder boundary = No

Allow partial last cylinder = Yes

Dump = No

Once the options were set, "Ok" was selected, followed by the selection of "Analyse". The following menu then presented "Quick Search", which was selected to start the analysis. The Testdisk Quick Search was started at 10:33 am Friday May 8, 2009 and completed sometime that weekend. Upon arriving at work 8:00 am the following Monday, May 11, 2009, Testdisk had completed its Quick Search run and was requesting to execute a "Deeper Search". The Deep Search was executed at 8:04 am Monday morning and took approximately 3.75[3] hours to complete.

The Testdisk program was run using the following commands from its own terminal window:

```
(24)        $ cd /media/sdb1

(25)        $ /opt/testdisk/testdisk_static /log /media/sdb1/suspect.dd
```

Command (25) also generated a log file that was stored as file */media/sdb1/testdisk.log*.

Upon completion of the tool, no valid or recognizable partition could be found anywhere on the disk image file. Therefore, at this point, it is safe to conclude that either the disk image file contains no partition signature of any sort or that if it exists, it is entirely proprietary in nature and undetectable by any tool unaware of its specific data structures.

[3] While Testdisk was conducting a thorough disk analysis for lost or damaged partitions, an in-depth Photorec data recovery analysis was underway, thereby significantly slowing down the process by consuming disk I/O bandwidth that would have otherwise been used exclusively by Testdisk.

3.2.2 Photorec

Photorec software is part of the Testdisk suite and is at the same version as Testdisk, version 6.11.3. The tool currently supports about 180 different file types. Photorec was run against the disk image file */media/sdb1/suspect.dd*. The tool was acquired in its precompiled form for Linux. The Photorec software was also located in */opt/testdisk*, the same location as the Testdisk program.

Photorec and Testdisk are relatively simple tools to use and provide simple and intuitive text-based interfaces that are easy to work with. Once the software was run against the disk image file, the program presented the "Proceed" option to the operator. This option was selected and then followed by "None" and "Options" to set the appropriate data and file search options. The tool's configurable options were set to:

Paranoid = Yes (Brute force enabled)

Allow partial last cylinder = Yes

Keep corrupted files = Yes

Expert mode = Yes

Low memory = No

Once the configurable options were set, "Quit" was selected, followed by "File Opt" to set the various file types available for recovery. All available file types were selected (most were already selected by default but some were left out) and when completed, "Quit" was selected. At this time, the previous menu returned at which time the "Search" option was selected. The ensuing menu then presented two sets of filesystems; the second option, "Other", was chosen, at which time the program began its search for recoverable data and files. However, this search, seen on the menu's display as "Pass 0", was actually a preliminary search, similar to Testdisk's Quick Search.

Pass 0 was started at 10:38 am Friday, May 8, 2009 and completed sometime over the weekend. Upon arriving at work the following Monday, Photorec had completed its Quick Search (Pass 0) run and was requesting to execute "Try to unformat a FAT filesystem (Y/N)." "N" was given as a response, at which time a series of various search block sizes were presented. A block size of "512" was selected followed by Enter. This subsequent processing took approximately four[4] hours to complete its in-depth scans.

The Photorec program was run using the following commands from its own terminal window:

(26) `$ cd /media/sdb1`

(27) `$ /opt/testdisk/photorec_static  /log  /d /tmp/photorec  /media/sdb1/suspect.dd`

[4] While Photorec was conducting a thorough disk analysis for lost or damaged partitions, an in-depth Testdisk data recovery analysis was underway thereby significantly slowing down the process by consuming disk I/O bandwidth that would have otherwise been used exclusively by Photorec.

Command (27) also generated a log file that was stored as file */media/sdb1/photorec.log*. All recovered files were extracted to directory */tmp/photorec.1* (the *.1* extension is automatically appended by the Photorec program to the destination directory). A total of 48 files were recovered and all were found to be dBase database files. All of the recovered files, upon closer inspection using OpenOffice 3.0.1 (which could read Dbase files), were determined to be corrupt.

3.2.3 Foremost

Foremost is an advanced file carving tool that, according to its source code files, recognizes 32 specific file formats, all of which are highly popular, including various office[5] and multimedia-based files. However, the tool's configuration file, *foremost.conf*, can be used to provide additional file signature capabilities to the tool. The current version used was 1.5.6 and was available only as source code. The software was easily compiled[6] and installed to */opt/foremost*. For the program to work, it is important that the default configuration file provided with the source code be copied to */opt/foremost* as well.

The software tool was run twice, once to detect and extract against the tool's default built-in file signatures and a second time based on the file signatures provided in the configuration file. However, by default, the configuration file is entirely commented out. Nevertheless, even if the configuration file is not providing any file signatures to the tool, its location must still be specified from the command line when running the program. Modifying its configuration file requires additional work, although it is relatively straightforward to enact using the following commands:

```
(28)        $ cd /opt/foremost

(29)        $ cp foremost.conf  foremost.conf.bak

(30)        $ vim foremost.conf
```

It is important to make a backup copy of the *foremost.conf* configuration file prior making any changes to it. Edit the original configuration file *foremost.conf* and remove all appropriate comment fields placed in front of the various file signatures/search patterns. When finished editing the file, save the file (a copy of the modified configuration file, *foremost.conf*, has been provided in Annex A.1). The configuration file backup with the original configuration settings, *foremost.conf.bak*, is used with the first instance of Foremost (see command (32) below) and should be considered as the default configuration file. However, the modified configuration file, *foremost.conf*, is used with the second instance of Foremost (see command (33) below).

Although Foremost is highly configurable as it can support most user-specified signatures, it has a practical limit to the number of signatures it can actively support. As determined through trial and error, the current version of the tool can support a maximum of 34 file signatures at a time. If a user requires additional file signatures, then the tool will have to be run another time with a new configuration file reflecting the required desired file signatures. The currently configured file signatures used with the tool for the second run of Foremost can be found in Annex A.1. Looking

[5] This includes both Microsoft Office and OpenOffice.

[6] It is expected that the reader is able to compile basic software packages under Linux.

at this example, the reader can easily discern which signatures were left commented out and which would be loaded and used by the program.

The following two iterations of the Foremost program were each run from its own terminal window and were executed as follows:

```
(31)        $ cd /media/sdb1

(32)        [Instance 1]    $ /opt/foremost/foremost -v -t all -c /opt/foremost/foremost.conf.bak  -o /tmp/tmp1  -i suspect.dd >foremost1.log&>foremost1.log &

(33)        [Instance 2]    $ /opt/foremost/foremost  -v  -c /opt/foremost/foremost.conf  -o /tmp/tmp2  -i suspect.dd >foremost2.log&>foremost2.log &
```

The two executed instances of the Foremost program were initiated at 10:41 and 10:44 am Friday, May 8, 2009, respectively. They both completed sometime over the weekend and according to their log files, they finished only a few minutes apart from one another. The log files, *foremost1.log* and *foremost2.log*, were saved to */tmp/tmp1* and */tmp/tmp2*, respectively. The first instance of Foremost, relying only on the tool's default built-in file signatures while its specified configuration file was left entirely commented out, supported 31 individual file types, as determined from the source code. The second instance of Foremost, based entirely on the newly modified configuration file, was able to recognize slightly more file types than the first instance of the tool; specifically 34 file types.

The first instance of Foremost that ran using its built-in file signatures did not succeed in finding or recovering any files or data. However, the second instance, using the modified configuration file, succeeded in finding the following file types:

- JPEG (*.jpg): 13,860 files

  -> Analysed using XnView version 1.96.1
- MP3 (*.mp3): 53 files

  -> Analysed using Windows Media Player 9
- MPEG (*.mpg): 43 files

  -> Analysed using XnView 1.96.1 and Windows Media Player 9 to view files
- PGP (*.pgp): 22,623 files

  -> Not possible to examine files as true PGP files would be encrypted
- PNG (*.png): 23 files

  -> Analysed using XnView 1.96.1 to view files
- WordPerfect (*.wpc): 4 files

  -> Analysed using OpenOffice 3.0.1 and Microsoft Office 2003 to open files

With the exception of the PGP files, which are impossible to directly analyze due to their "encrypted" content, all the other file types were analyzed using the aforementioned tools throughout May 12 and 13, 2009. All of the variously examined files were found to be corrupt. As for the large selection of PGP files, a small random selection of 283 (or 1.25%) PGP files were examined with a hex/text editor to verify that these files in fact contained no useful or discernible information. The vast majority of the examined PGP files were only several kilobytes in size, thereby making analysis less painstaking. Although no hard conclusion can be made about the state of these PGP files since their data is purportedly encrypted, then viewing them through a hex/text editor should reveal very little about them or their internal structures. Thus, it is logical to conclude that they too contain no useful information or data.

3.2.4 Recoverjpeg

The program Recoverjpeg is a highly specialized tool whose sole purpose is to find and extract potential JPEG images from disk images and devices. It is generally very good at finding JPEG-based images since it does absolutely nothing else. The tool is not particularly prone to false positives, unlike the other previously used tools. Moreover, the tool is straightforward. It was executed as follows in its own terminal window:

```
(34)        $ cd /media/sdb1

(35)        $ /opt/recoverjpeg/recoverjpeg -v suspect.dd >recoverjpeg.log&>recoverjpeg.log
&
```

Recoverjpeg version 1.1.4 was used and found no evidence of any possibly recoverable JPEG images. A log file was generated by the tool, which upon analysis clearly indicated that nothing had been found. The only downside to this tool is that it does not have the ability to allow the user to specify where to save recovered files. Recovered files are saved to the current working directory. The program was launched at mid-morning May 8, 2009, and completed in the early morning of May 9, 2009.

3.2.5 Scalpel

Users who have used Foremost before are likely to have noticed many similarities between Foremost and Scalpel. This is because Scalpel is actually based on Foremost version 0.69. Scalpel offers additional file carving capabilities that Foremost does not. Contrary to Foremost, Scalpel does not support any specific file type for recovery. Therefore, in order to successfully recover data, at least one valid file type signature must be specified in its configuration file *scalpel.conf*. Moreover, its configuration does not have the same limitations as Foremost. It is possible to specify far more file type signatures than Foremost could support. Scalpel version 1.60 was used for this data recovery. It was run from its own terminal window as follows:

```
(36)        $ cd /media/sdb1

(37)        $ cp /etc/scalpel/scalpel.conf /media/sdb1

(38)        $ vim scalpel.conf
```

```
(39) $ scalpel -b -c scalpel.conf -o /tmp/scalpel -v suspect.dd >scalpel.log&>scalpel.log
```

For the tool to work it is necessary that its configuration file *scalpel.conf* contain at least one valid file type entry. A copy of the configuration file used herein can be found in Annex A.2. The tool was launched at 10:55 am Friday, May 8, 2009 and completed sometime over the weekend. It succeeded in finding the following file types:

- MPEG (*.mpg): 11 files

  -> Used XnView 1.96.1 and Windows Media Player 9 to view files
- PGP (*.pgp): 22,747 files

  -> Not possible to examine files as true PGP files would be encrypted
- RPM (*.rpm): 15,550 files

  -> Used Linux *rpm* tool to assess file integrity
- WordPerfect (*.wpc): 4 files

  -> Used OpenOffice 3.0.1 to open files / also tried Microsoft Office 2003

With the exception of the PGP files, which are impossible to analyze due to their encrypted content, all the other files were analyzed using their aforementioned respective tools throughout the days of May 13 and 14, 2009. All the files were found to be corrupt. A small random selection of 284 (or 1.25%) PGP files were examined with a hex/text editor to verify that these files in fact contained no useful or discernible information. The vast majority of the examined PGP files were only several kilobytes in size, thereby making analysis less painstaking.

All the files were found to be corrupt, although no hard conclusion can be made about the state of the PGP files, since their data is supposedly encrypted and viewing them through a hex/text editor revealed very little about them or their internal structures. It is therefore justifiable to assume that they too contain no useful information or data.

Although a large number of RPM files were recovered, it was possible to analyze them all using only the command shell. The Linux rpm command can accept various file name patterns and in using these patterns, analysis of these files was only a matter of specifying several hundred at a time so that their errant output would not overflow the terminal window's line buffers.

Analysis of these files took place on May 14 and May 15 in order to accurately determine their underlying nature.

3.2.6 Magicrescue

Magicrescue is the final Linux-based file carving and extraction tool examined herein. The tool works by detecting potential file signatures and extracting them based on a series of rules called recipes. These recipes enable the program to perform specific actions and apply certain data validation techniques in order to distinguish valid data files from similar-appearing nonsensical data and save valid files to disk. The program used was at version 1.1.6 and currently supports 16 highly common file types. The program was run as follows from its own terminal window:

```
$ cd /media/sdb1

$ /opt/magicrescue/magicrescue -d /tmp/magicrescue -r /opt/magicrescue/recipes suspect.dd >magicrescue.log&>magicrescue.log &
```

Here, successfully extracted data is saved to directory */tmp/magicrescue* based upon the default recipes that are loaded from */opt/magicrescue/recipes*. The program found three MP3 files that were played back using both Windows Media Player 9 and Linux-based Rhythmbox Music Player (under Ubuntu Linux). The three MP3 files were found to be corrupt and unplayable. The program was launched during the late morning of May 8, 2009 and completed the evening of May 9, 2009. The MP3 files were tested during the late morning of May 11, 2009.

3.3 Additional Notes

All the aforementioned Linux forensic tools had to be downloaded from a different network-connected workstation, burned to optical disc before being transferred to the forensic workstation host system, and then scanned using the host system's AVG antivirus. Only upon a successful virus scanning was the optical disc mounted under Ubuntu Linux. All the various software tools, dependencies and updates had to be manually downloaded from the network-enabled download workstation. Although not a difficult task, it was time consuming to verify and satisfy software dependencies, although there were few to satisfy.

The following is a list of the number of recoverable file types supported by the various aforementioned data recovery programs:

- Testdisk: 30+ partition types
- Photorec: about 180 file types
- Foremost: 32 file types supported by default with ability to search for user-specified signatures through use of configuration file
- Recoverjpeg: Jpeg file formats only
- Scalpel: by default the tool does not support any specific file type of signature as it requires the user to specify file signatures through its configuration file
- Magicrescue: 16

Although not explicitly stated, each program, where applicable, was configured to detect the maximum possible number of file types. Some programs (e.g. Photorec) by default had most but not all selected, while other tools by default supported all their default file types for recovery (e.g. Foremost). Others by default supported no specific file type and required the user to specify which ones through configuration files (e.g. Scalpel).

Analysis of which files to analyse when there too many to examine manually within a reasonable period of time were conducted using a statistically sound sampling-based methodology as previously examined in Section 2.4.

3.4 Conclusion

Despite the inherent capabilities of the various Linux and UNIX data recovery tools examined herein, none was able to extract any useful information, files or data from the suspect disk-based image file. Although all the recovered files were successfully analyzed, they were found to be corrupt and unusable. Contrary to the Windows data recovery portion of this memorandum, all the recovered file types were analyzable using the current software available to the operator. However, based on the results from the analysis of the variously recovered files, it is very unlikely that any of them contained useful or useable information or data.

It was hoped at the outset of this analysis that using various Windows tools from differing vendors would have been able to provide some meaningful results. When those tools failed to detect any useful data or information, it was then hoped that various advanced Linux and UNIX data recovery tools would successfully recover some useful data or information. However, based on the results obtained, it can be stated that none of the files recovered have anything to do with a DVR-related device, as per the information received from the external agency. In addition, all of the analyzed files appear to be corrupt and entirely unusable. Furthermore, with the exception of the Testdisk tool, all the Linux data recovery tools are advanced file-carving tools, similar in technology and capability to their Windows counterparts.

Therefore, based on these results, it would seem prudent to state that analyzing an unknown filesystem such as this one, with unknown file structures, a manual data extraction may be the only likely avenue for successfully recovering data, information, or files.

4 Wrapping up

Upon completion of the all the various data recovery operations, both the generated and copied disk image files for both Windows and Linux-based, including the secondary target drive, were re-MD5 hashed and compared against the MD5 hash of the original suspect disk. This was conducted in order to ensure that none of the disk image files or target drive had become modified or compromised because of the data recovery tools used against them. After having rehashed all the disk image files and target drive, it was found that no detectable changes had been made to any of the files or drive. Thus, although no useful information or data could be retrieved from the images or target drive, at least it was not a result of data modification or corruption.

The Linux VMware machine was permanently turned off and its VMware data files, including the virtual machine's storage VMDK file, were wiped using a 7-pattern DoD-compliant wipe to ensure that no data remained on the system. Furthermore, the Windows host system's swap space was also wiped using the same DoD-compliant wipe. Data wiping was conducted using BCWipe version 3. All Windows-based data recovery tools were uninstalled from the Windows forensic workstation and all potentially undeleted application directories and temporary storage locations were deleted. Furthermore, the system's history, application, security and event logs were deleted. Then, using BCWipe, all recently used files were wiped using the aforementioned DoD-compliant wipe. Finally, the system's free space was also wiped followed by a wiping of system swap space using the DoD-compliant wipe in order to ensure that the system could be used again for future forensic analyses.

Finally, the secondary target drive was wiped from a Helix Live CD and using the tool *shred*, it was wiped with 25 patterns. However, the primary target drive was put away for safekeeping in order completing the second part of this study, which examines using pattern matching techniques to recover data. This is examined in an upcoming report.

5 Conclusion

The objective of this memorandum was to determine if it would be possible to recover files, data or information from a disk drive emanating from an unknown DVR security-recording device. However, the disk drive used in this experiment utilizes an obscure filesystem that stores unknown file structures. After subjecting the disk image file of the original suspect disk to a full battery of data recovery tests, absolutely nothing of any value has been recovered or extracted from this disk device. Although these tools are very capable in their own right and have been able to recover data from other computer systems, they have had no effect here.

Although it is important to avoid generalizations, it appears that standard data recovery tools, as advanced as they are, are entirely incapable of recovery this disk's data. Furthermore, after having used such a wide array of tools against the disk, it is unlikely that any tool, commercial or open source in nature, would be able to. The aforementioned tools represent a wide gamut of data recovery technologies and as such should be highly representative of most present-day tool capabilities.

Unfortunately, data recovery tools are capable of carrying their tasks only when presented with certain known factors about the data to recover. For example, an intact filesystem is always the preferred method of recovering data. However, using file structures and signatures is the least preferred method and does not always yield desirable results. Furthermore, the latter type of recovery methods is highly prone to false positives. Some of the data recovery tools were particularly prone to false positives, while others did not manage to recover even one false positive.

The problem at hand for the disk image is that the data stored on this disk is using a filesystem that is both obscure, as it is entirely undocumented and proprietary in nature. For hardware designers, especially for high-end security devices, the tendency is towards proprietary technology. Furthermore, the files that do exist are entirely unrecognizable due to their very specific file headers and data structures. To further complicate matters, the video streams represented by files are in fact not only encrypted, but also watermarked, making their recovery all the more problematic. However, a preliminary pattern matching data recovery analysis has already yielded tangible results that will be explored in a follow-up study.

Today's data recovery tools place much emphasis on recovering data based on file headers, signatures and structures. This approach works well when attempting to recover commonly used data and files typical of the workplace or an individual's home. When the tools are placed up against an entirely unknown disk containing obscure data, the tools fail, not specifically due to flaws or design errors, but rather from their underlying function, which is to recover only what they can recognize.

This memorandum has served as a useful example of how to perform a data recovery analysis against an unknown disk with obscure contents. Furthermore, memorandum it has clearly demonstrated the limits of modern data recovery tools and the manner in which they work. In the computer forensics field, there is an overreliance on automated data recovery tools which, as seen here, clearly have their limitations. It is therefore important for computer forensic investigators to be able to carry out an investigation using manually operated analytical tools. The aforementioned follow-up study will specifically examine the use of such tools.

References

[1] Wikipedia. Entry: Sampling (statistics). Online encyclopaedic entry. Wikimedia Inc. June 2009. http://en.wikipedia.org/wiki/Sampling_(statistics)#Simple_random_sampling.

This page intentionally left blank.

Annex A Configuration files

A.1 Foremost configuration file

The following is the configuration file for Foremost, found in Subsection 3.2.3:

```
#
# Foremost configuration file
#---------------------------------------------------------------------
# Note the foremost configuration file is provided to support formats which
# don't have built-in extraction functions.  If the format is built-in to foremost
# simply run foremost with -t <suffix> and provide the format you wish to extract.
#
# The configuration file is used to control what types of files foremost
# searches for. A sample configuration file, foremost.conf, is included with
# this distribution. For each file type, the configuration file describes
# the file's extension, whether the header and footer are case sensitive,
# the maximum file size, and the header and footer for the file. The footer
# field is optional, but header, size, case sensitivity, and extension are
# not!
#
# Any line that begins with a '#' is considered a comment and ignored. Thus,
# to skip a file type just put a '#' at the beginning of that line
#

# Headers and footers are decoded before use. To specify a value in
# hexadecimal use \x[0-f][0-f], and for octal use \[0-3][0-7][0-7].  Spaces
# can be represented by \s. Example: "\x4F\123\I\sCCI" decodes to "OSI CCI".
#
# To match any single character (aka a wildcard) use a '?'. If you need to
# search for the '?' character, you will need to change the 'wildcard' line
# *and* every occurrence of the old wildcard character in the configuration
# file. Don't forget those hex and octal values! '?' is equal to 0x3f and
# \063.
#
# If you would like to extract files without an extension enter the value
# "NONE" in the extension column (note: you can change the value of this
# "no suffix" flag by setting the variable FOREMOST_NOEXTENSION_SUFFIX
# in foremost.h and recompiling).
#
# The ASCII option will extract all ASCII printable characters before and after
# the keyword provided.
#
# The NEXT keyword after a footer instructs foremost to search forwards for data
# that starts with the header provided and terminates or is followed by data in
# the footer -- the footer data is not included in the output.  The data in the
# footer, when used with the NEXT keyword effectively allows you to search for
# data that you know for sure should not be in the output file.  This method for
# example, lets you search for two 'starting' headers in a document that doesn't
# have a good ending footer and you can't say exactly what the footer is, but
# you know if you see another header, that should end the search and an output
# file should be written.

# To redefine the wildcard character, change the setting below and all
# occurances in the formost.conf file.
#
#wildcard  ?
#
#               case    size    header                  footer
#extension   sensitive
#
#---------------------------------------------------------------------
# EXAMPLE WITH NO SUFFIX
#---------------------------------------------------------------------
#
# Here is an example of how to use the no extension option. Any files
# containing the string "FOREMOST" would be extracted to a file without
# an extension (eg: 00000000,00000001)
#    NONE     y    1000     FOREMOST
#
#---------------------------------------------------------------------
# GRAPHICS FILES
#---------------------------------------------------------------------
#
#
# AOL ART files
art     y       150000 \x4a\x47\x04\x0e         \xcf\xc7\xcb
art     y       150000 \x4a\x47\x03\x0e         \xd0\xcb\x00\x00
#
# GIF and JPG files (very common)
#       (NOTE THESE FORMATS HAVE BUILTIN EXTRACTION FUNCTION)
gif     y       155000000       \x47\x49\x46\x38\x37\x61        \x00\x3b
gif     y       155000000       \x47\x49\x46\x38\x39\x61        \x00\x00\x3b
jpg     y       20000000        \xff\xd8\xff\xe0\x00\x10  \xff\xd9
jpg     y       20000000        \xff\xd8\xff\xe1 \xff\xd9
jpg     y       20000000        \xff\xd8  \xff\xd9
#
# PNG   (used in web pages)
#       (NOTE THIS FORMAT HAS A BUILTIN EXTRACTION FUNCTION)
png     y       200000 \x50\x4e\x47?   \xff\xfc\xfd\xfe
#
#
# BMP
#       (NOTE THIS FORMAT HAS A BUILTIN EXTRACTION FUNCTION)
bmp     y       100000 BM??\x00\x00\x00
#
# TIF
tif     y       200000000       \x49\x49\x2a\x00
#
#---------------------------------------------------------------------
# ANIMATION FILES
#---------------------------------------------------------------------
#
```

```
# AVI (Windows animation and DiVX/MPEG-4 movies)
#         (NOTE THIS FORMAT HAS A BUILTIN EXTRACTION FUNCTION)
avi       y         4000000 RIFF????AVI
#
# Apple Quicktime
#         (NOTE THIS FORMAT HAS A BUILTIN EXTRACTION FUNCTION)
mov       y         4000000             ????????\x6d\x6f\x6f\x76
mov       y         4000000             ????????\x6d\x64\x61\x74
#
# MPEG Video
mpg       y         4000000             mpg       eof
mpg       y         20000000 \x00\x00\x01\xba      \x00\x00\x01\xb9
mpg      y          20000000 \x00\x00\x01\xb3       \x00\x00\x01\xb7
#
# Macromedia Flash
#         fws       y         4000000             FWS
#
#--------------------------------------------------------------------
# MICROSOFT OFFICE
#--------------------------------------------------------------------
#
# Word documents
#         (NOTE THIS FORMAT HAS A BUILTIN EXTRACTION FUNCTION)
doc       y         12500000  \xd0\xcf\x11\xe0\xa1\xb1
#
# Outlook files
pst       y         400000000 \x21\x42\x4e\xa5\x6f\xb5\xa6
ost       y         400000000 \x21\x42\x44\x4e
#
# Outlook Express
dbx       y         4000000             \xcf\xad\x12\xfe\xc5\xfd\x74\x6f
idx       y         4000000             \x4a\x4d\x46\x39
mbx       y         4000000             \x4a\x4d\x46\x36
#
#--------------------------------------------------------------------
# WORDPERFECT
#--------------------------------------------------------------------
#
wpc       y         100000 ?WPC
#
#--------------------------------------------------------------------
# HTML              (NOTE THIS FORMAT HAS A BUILTIN EXTRACTION FUNCTION)
#--------------------------------------------------------------------
#
htm       n         50000   <html                         </html>
#
#--------------------------------------------------------------------
# ADOBE PDF  (NOTE THIS FORMAT HAS A BUILTIN EXTRACTION FUNCTION)
#--------------------------------------------------------------------
#
pdf       y         5000000             %PDF-  %EOF
#
#
```

```
#------------------------------------------------------------------------
# AOL (AMERICA ONLINE)
#------------------------------------------------------------------------
#
# AOL Mailbox
mail       y          500000  \x41\x4f\x4c\x56\x4d
#
#
#
#------------------------------------------------------------------------
# PGP (PRETTY GOOD PRIVACY)
#------------------------------------------------------------------------
#
# PGP Disk Files
pgd        y          500000 \x50\x47\x50\x64\x4d\x41\x49\x4e\x60\x01
#
# Public Key Ring
pgp        y          100000 \x99\x00
# Security Ring
pgp        y          100000 \x95\x01
pgp        y          100000 \x95\x00
# Encrypted Data or ASCII armored keys
pgp        y          100000 \xa6\x00
# (there should be a trailer for this...)
txt        y          100000 -----BEGIN\040PGP
#
#
#------------------------------------------------------------------------
# RPM (Linux package format)
#------------------------------------------------------------------------
#rpm     y          1000000          \xed\xab
#
#
#------------------------------------------------------------------------
# SOUND FILES
#------------------------------------------------------------------------
#          (NOTE THIS FORMAT HAS A BUILTIN EXTRACTION FUNCTION)
wav      y           200000 RIFF????WAVE
#
# Real Audio Files
ra         y          1000000          \x2e\x72\x61\xfd
ra         y          1000000          .RMF
#
asf     y      8000000      \x30\x26\xB2\x75\x8E\x66\xCF\x11\xA6\xD9\x00\xAA\x00\x62\xCE\x6C
#
wmv     y      20000000 \x30\x26\xB2\x75\x8E\x66\xCF\x11\xA6\xD9\x00\xAA\x00\x62\xCE\x6C
#
wma     y      8000000  \x30\x26\xB2\x75    \x00\x00\x00\xFF
#
wma     y      8000000  \x30\x26\xB2\x75    \x52\x9A\x12\x46
#
mp3     y          8000000 \xFF\xFB??\x44\x00\x00
mp3     y          8000000 \x57\x41\x56\45          \x00\x00\xFF\
mp3     y           8000000 \xFF\xFB\xD0\           \xD1\x35\x51\xCC\
mp3     y           8000000 \x49\x44\x33\
mp3     y           8000000 \x4C\x41\x4D\x45\
#---------------------------------------------------------------------
# WINDOWS REGISTRY FILES
#---------------------------------------------------------------------
#
# Windows NT registry
#dat      y          4000000           regf
# Windows 95 registry
#dat      y          4000000           CREG
#
#lnk     y           5000     \x4C\x00\x00\x00\x01\x14\x02\x00\x00\x00\x00\x00\xC0\x00\x00
#chm     y          100000 \x49\x54\x53\x46\x03\x00\x00\x00\x60\x00\x00\x00\x01\x00\x00
#cookie  n          4096    id=
#rdp     y          4096
          \xFF\xFE\x73\x00\x63\x00\x72\x00\x65\x00\x65\x00\x6E\x00\x20\x00\x6D
#
#---------------------------------------------------------------------
# MISCELLANEOUS
#---------------------------------------------------------------------
#         (NOTE THIS FORMAT HAS BUILTIN EXTRACTION FUNCTION)
zip       y          10000000          PK\x03\x04       \x3c\xac
#         (NOTE THIS FORMAT HAS BUILTIN EXTRACTION FUNCTION)
rar       y          10000000          Rar!
#
java      y          1000000           \xca\xfe\xba\xbe
#
cpp       y          20000   #include           #include           ASCII
#---------------------------------------------------------------------
# ScanSoft PaperPort "Max" files
#---------------------------------------------------------------------
max   y     1000000    \x56\x69\x47\x46\x6b\x1a\x00\x00\x00\x00   \x00\x00\x05\x80\x00\x00
#---------------------------------------------------------------------
# PINs Password Manager program
#---------------------------------------------------------------------
pins  y     8000     \x50\x49\x4e\x53\x20\x34\x2e\x32\x30\x0d
```

A.2 Scalpel configuration file

The following is the configuration file for Scalpel, found in Subsection 3.2.5:

```
# Scalpel configuration file

# This configuration file controls the
# types and sizes of files that are carved by Scalpel.  Currently,
# Scalpel can read Foremost 0.69 configuration files, but Scalpel
# configuration files may not be backwards-compatible with Foremost.
# In particular, maximum file carve size under Foremost 0.69 is 4 GB,
# while in the current version of Scalpel, it's 16 EB (16 exabytes).

# For each file type, the configuration file
# describes the file's extension, whether the header and footer are
# case sensitive, the maximum file size, and the header and footer for
# the file. The footer field is optional, but header, size, case
# sensitivity, and extension are required.  Any line that begins with a
# '#' is considered a comment and ignored. Thus, to skip a file type
# just put a '#' at the beginning of that line

# Headers and footers are decoded before use. To specify a value in
# hexadecimal use \x[0-f][0-f] and for octal use \[0-3][0-7][0-7].
# Spaces can be represented by \s. Example: "\x4F\123\I\sCCI" decodes
# to "OSI CCI".  # To match any single character (aka a wildcard) use
# a '?'. If you need to search for the '?' character, you will need to
# change the 'wildcard' line *and* every occurrence of the old
# wildcard character in the configuration file. '
#
# Note: ?' is equal to 0x3f and \063.
#
# If you want files carved without filename extensions,
# use "NONE" in the extension column.

# The REVERSE keyword after a footer causes a search
# backwards starting from [size] bytes beyond the location of the header
# This is useful for files like PDFs that may contain multiple copies of
# the footer throughout the file.  When using the REVERSE keyword you will
# extract bytes from the header to the LAST occurence of the footer (and
# including the footer in the carved file).
#
# The NEXT keyword after a footer results in file carves that
# include the header and all data BEFORE the first occurence of the
# footer (the footer is not included in the carved file).  If no
# occurrence of the footer is discovered within maximum carve size bytes
# from the header, then a block of the disk image including the header
# and with length equal to the maximum carve size is carved.  Use NEXT
# when there is no definitive footer for a file type, but you know which
# data should NOT be included in a carved file--e.g., the beginning of
# a subsequent file of the same type.
#
# FORWARD_NEXT is the default carve type and this keyword may be
# included after the footer, but is not required.  For FORWARD_NEXT
# carves, a block of data including the header and the first footer
# (within the maximum carve size) are carved.  If no footer appears
# after the header within the maximum carve size, then no carving is
# performed UNLESS the -b command line option is supplied.  In this case,
# a block of max carve size bytes, including the header, is carved and a
# notation is made in the Scalpel log that the file was chopped.

# To redefine the wildcard character, change the setting below and all
# occurences in the formost.conf file.
#
#wildcard  ?

#               case    size    header                  footer
#extension  sensitive
#
#--------------------------------------------------------------------
# EXAMPLE WITH NO SUFFIX
#--------------------------------------------------------------------
#
# Here is an example of how to use the no extension option. Any files
# beginning with the string "FOREMOST" are carved and no file extensions
# are used. No footer is defined and the max carve size is 1000 bytes.
#
#     NONE     y      1000     FOREMOST
#
#--------------------------------------------------------------------
# GRAPHICS FILES
#--------------------------------------------------------------------
#
#
# AOL ART files
        art     y       150000 \x4a\x47\x04\x0e         \xcf\xc7\xcb
        art     y       150000 \x4a\x47\x03\x0e         \xd0\xcb\x00\x00
#
# GIF and JPG files (very common)
        gif     y       5000000                 \x47\x49\x46\x38\x37\x61        \x00\x3b
        gif     y       5000000                 \x47\x49\x46\x38\x39\x61        \x00\x3b
        jpg     y       200000000       \xff\xd8\xff\xe0\x00\x10  \xff\xd9
#
#
# PNG
        png     y       20000000        \x50\x4e\x47?   \xff\xfc\xfd\xfe
#
#
# BMP  (used by MSWindows, use only if you have reason to think there are
#       BMP files worth digging for. This often kicks back a lot of false
#       positives
#
        bmp     y       100000 BM??\x00\x00\x00
#
# TIFF
        tif     y       200000000       \x49\x49\x2a\x00
```

```
# TIFF
        tif     y       200000000       \x4D\x4D\x00\x2A
#
#---------------------------------------------------------------------
# ANIMATION FILES
#---------------------------------------------------------------------
#
# AVI (Windows animation and DiVX/MPEG-4 movies)
        avi     y       50000000 RIFF????AVI
#
# Apple Quicktime
#   These needles are based on the file command's magic.  I don't
#   recommend uncommenting the 4th and 5th Quicktime needles unless
#   you're sure you need to, because they generate HUGE numbers of
#   false positives.
#
        mov     y       10000000        ????moov
        mov     y       10000000        ????mdat
        mov     y       10000000        ????widev
        mov     y       10000000        ????skip
        mov     y       10000000        ????free
        mov     y       10000000        ????idsc
        mov     y       10000000        ????pckg
#
# MPEG Video
        mpg     y       50000000        \x00\x00\x01\xba        \x00\x00\x01\xb9
        mpg    y        50000000        \x00\x00\x01\xb3        \x00\x00\x01\xb7
#
# Macromedia Flash
        fws     y       4000000         FWS
#
#---------------------------------------------------------------------
# MICROSOFT OFFICE
#---------------------------------------------------------------------
#
# Word documents
#
#
        doc     y       10000000  \xd0\xcf\x11\xe0\xa1\xb1\x1a\xe1\x00\x00
\xd0\xcf\x11\xe0\xa1\xb1\x1a\xe1\x00\x00 NEXT
        doc     y       10000000  \xd0\xcf\x11\xe0\xa1\xb1
#
# Outlook files
        pst     y       500000000       \x21\x42\x4e\xa5\x6f\xb5\xa6
        ost     y       500000000       \x21\x42\x44\x4e
#
# Outlook Express
        dbx     y       10000000        \xcf\xad\x12\xfe\xc5\xfd\x74\x6f
        idx     y       10000000        \x4a\x4d\x46\x39
        mbx     y       10000000        \x4a\x4d\x46\x36
#
#---------------------------------------------------------------------
# WORDPERFECT
```

```
#-------------------------------------------------------------------
#
        wpc     y       1000000         ?WPC
#
#-------------------------------------------------------------------
# HTML
#-------------------------------------------------------------------
#
        htm     n       50000   <html                   </html>
#
#-------------------------------------------------------------------
# ADOBE PDF
#-------------------------------------------------------------------
#
        pdf     y       5000000         %PDF  %EOF\x0d        REVERSE
        pdf     y       5000000         %PDF  %EOF\x0a        REVERSE
#
#-------------------------------------------------------------------
# AOL (AMERICA ONLINE)
#-------------------------------------------------------------------
#
# AOL Mailbox
        mail    y       500000  \x41\x4f\x4c\x56\x4d
#
#
#
#-------------------------------------------------------------------
# PGP (PRETTY GOOD PRIVACY)
#-------------------------------------------------------------------
#
# PGP Disk Files
        pgd     y       500000 \x50\x47\x50\x64\x4d\x41\x49\x4e\x60\x01
#
# Public Key Ring
        pgp     y       100000 \x99\x00
# Security Ring
        pgp     y       100000 \x95\x01
        pgp     y       100000 \x95\x00
# Encrypted Data or ASCII armored keys
        pgp     y       100000 \xa6\x00
# (there should be a trailer for this...)
        txt     y       100000 -----BEGIN\040PGP
#
#
#-------------------------------------------------------------------
# RPM (Linux package format)
#-------------------------------------------------------------------
        rpm     y       1000000         \xed\xab
#
#
#-------------------------------------------------------------------
# SOUND FILES
#-------------------------------------------------------------------
```

```
#
        wav     y           200000 RIFF????WAVE
#
# Real Audio Files
        ra        y         1000000         \x2e\x72\x61\xfd
        ra        y         1000000         .RMF
#
#----------------------------------------------------------------------
# WINDOWS REGISTRY FILES
#----------------------------------------------------------------------
#
# Windows NT registry
        dat       y         4000000         regf
# Windows 95 registry
        dat       y         4000000         CREG
#
#
#----------------------------------------------------------------------
# MISCELLANEOUS
#----------------------------------------------------------------------
#
        zip       y         10000000        PK\x03\x04      \x3c\xac
#
        java      y         1000000         \xca\xfe\xba\xbe
#
#----------------------------------------------------------------------
# ScanSoft PaperPort "Max" files
#----------------------------------------------------------------------
    max   y    1000000    \x56\x69\x47\x46\x6b\x1a\x00\x00\x00\x00   \x00\x00\x05\x80\x00\x00
#----------------------------------------------------------------------
# PINs Password Manager program
#----------------------------------------------------------------------
    pins  y    8000     \x50\x49\x4e\x53\x20\x34\x2e\x32\x30\x0d
```

Bibliography

DiskInternals Research. DiskInternals Partition Recovery help file. Help file. DiskInternals Partition Recovery. 2009. http://www.diskinternals.com/partition-recovery.

Grenier, Christophe. Photorec man file. Man file. CGSecurity web site c/o Christophe Grenier. April 2009. http://www.cgsecurity.org/wiki/PhotoRec.

Grenier, Christophe. Testdisk man file. Man file. CGSecurity web site c/o Christophe Grenier. April 2009. http://www.cgsecurity.org/wiki/TestDisk.

Jensen, Jonas. Magicrescue man file. Man file. Magic Rescue web site c/o Jonas Jensen. February 2009. http://www.student.dtu.dk/~s042078/magicrescue/.

Kendall, Kris, and Jesse Kornblum. Foremost man file. Man file. Foremost web site c/o Sourceforge. May 2009. http://foremost.sourceforge.net/.

Kroll Ontrack Inc. Ontrack EasyRecovery Professional help file. Help file. Ontrack EasyRecovery Professional. 2009. http://www.ontrackdatarecovery.com/file-recovery-software.

Richard III, Golden G. Scalpel man file. Man file. Scalpel web site c/o Golden G. Richard III. December 2006. http://www.digitalforensicssolutions.com/Scalpel/.

R-tools Technology Inc. R-Studio help file. Help file. Help file R-Studio. 2009. http://www.r-studio.com.

Stellar Information Systems Ltd . Stellar Phoenix Photo Recovery help file. Help file. Stellar Phoenix Photo Recovery. 2009. http://www.stellarinfo.com/digital-media-recovery.htm.

Stellar Information Systems Ltd. Stellar Phoenix Windows Data Recovery help file. Help file. Stellar Phoenix Windows Data Recovery. 2009. http://www.stellarinfo.com/partition-recovery.htm.

Tardieu, Samuel. Recoverjpeg man file. Man file. Recoverjpeg web site c/o Samuel Tardieu. February 2009. http://www.rfc1149.net/devel/recoverjpeg.

List of symbols/abbreviations/acronyms/initialisms

10 k	10,000 or ten thousand
15 k	15,000 or fifteen thousand
3D	Three Dimensional
ARJ	Archived by Robert Jung
CAD	Computer Aided Design
CD	Compact Disc
CPU	Central Processing Unit
DoD	Department of Defense
DRDC	Defence Research & Development Canada
DVD	Digital Video Disc or Digital Versatile Disc
DVR	Digital Video Recorder
Ext2/Ext3	Second Extended Filesystem/Third Extended Filesystem
FAT32	File Allocation Table 32
FC6	Fedora Core 6
FTP	File Transfer Protocol
GB	Gigabyte (10^9 bytes)
GRUB	Grand Unified Boot Loader
HFS/HFS+	Hierarchal File System/Hierarchal File System+
HT	HyperThreading
IDE	Integrated Device Electronics
I/O	Input/Output
JPEG	Joint Photographic Experts Group
MB	Megabyte (10^6 bytes)
MD5	Message Digest algorithm 5
MP3	Moving Picture Experts Group-1 Audio Layer 3
MPEG	Moving Picture Experts Group
NTFS	NT File System
PC	Personal Computer
PE	Portable Executable
PGM	Portable Gray Map

PPM	Portable Pixmap
R/RW/RAM	Read/Read-Write/Random Access Memory
RAID	Redundant Array of Inexpensive Disks
RAM	Random Access Memory
RPM	Revolutions Per Minute
SATA	Serial Advanced Technology Attachment
SMP	Symmetric Multiprocessing
SCSI	Small Computer System Interface
TIFF	Tagged Image File Format
UFS1/UFS2	Unix File System 1/Unix File System 2
USB	Universal Serial Bus
VMDK	Virtual Machine Disk

This page intentionally left blank.

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Defence R&D Canada – Valcartier
2459 Pie-XI Blvd North
Quebec (Quebec)
G3J 1X5 Canada

2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.)

UNCLASSIFIED
(NON-CONTROLLED GOODS)
DMC A
REVIEW: GCEC JUNE 2010

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

File recovery and data extraction using automated data recovery tools: A balanced approach using Windows and Linux when working with an unknown disk image and filesystem

4. AUTHORS (last name, followed by initials – ranks, titles, etc. not to be used)

Carbone, R.

5. DATE OF PUBLICATION (Month and year of publication of document.)

January 2013

6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.)

66

6b. NO. OF REFS (Total cited in document.)

1

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Technical Memorandum

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

Defence R&D Canada – Valcartier
2459 Pie-XI Blvd North
Quebec (Quebec)
G3J 1X5 Canada

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)

9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)

DRDC Valcartier TM 2009-161

10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

Unlimited

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.))

Unlimited

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

(U) This memorandum is the direct result of the analysis of an unknown disk containing unknown data, files and filesystem. The disk was brought to an analysis team at DRDC Valcartier by an agency that desired to ascertain the research centre's capabilities for extracting and recovering unknown forensic data from an unknown disk and, if possible, automate the process. However, a thorough analysis using various Windows and Linux-based automated data and file recovery tools has led the author to determine that automated tools, regardless of the underlying system, are not yet up to this specific challenge. In addition, the author is of the opinion that fully automated disk recovery tools will never be entirely successful. Instead, the author has determined that a manual approach to data and file extraction will be necessary in order to recover any meaningful data or files from this disk's unknown filesystem. However, this memorandum will only examine the automated approach used by the various Windows and Linux tools. An additional follow-up study will specifically examine the required manual approach necessary for data recovery from an unknown disk using data pattern matching techniques and sector-by-sector analysis using known file signatures.

(U) Ce mémorandum est le résultat direct de l'analyse d'un disque inconnu contenant des données, des dossiers, et un système de fichiers obscur. Le disque a été apporté à une équipe d'analyse à DRDC Valcartier par une agence qui a désiré établir les capacités de centre de recherches pour extraire et en récupérant des données d'informatiques légales inconnues à partir d'un disque inconnu et, si possible, automatisez le processus. Cependant, une analyse complète utilisant divers outils de reconstitution de fichier et des données automatisées pour Windows et Linux a mené l'auteur à déterminer que les outils automatisés, indépendamment du système sous-jacent, ne sont pas encore jusqu'à ce défi spécifique. En outre, l'auteur est de l'opinion que les outils entièrement automatisés de récupération de disque ne seront jamais entièrement réussis. Au lieu de cela, l'auteur a déterminé qu'une approche manuelle aux données et à l'extraction de dossier sera nécessaire afin de récupérer tous les données ou dossiers indicatifs à partir de système de fichier obscur de ce disque. Cependant, ce mémorandum examinera seulement l'approche automatisée employée par les divers outils de Windows et de Linux. Une étude complémentaire examinera spécifiquement l'approche manuelle exigée nécessaire pour la récupération de données à partir d'un disque inconnu utilisant des techniques de configuration avec un modèle de données et l'analyse secteur par secteur utilisant les signatures connues de dossier.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Carving; Computer forensics; Data carving; Data Extraction; Data Recovery; Digital forensics; File carving; File Extraction; File Recovery; Filesystems; Forensics; FOSS; Linux; Open source software; Unknown Filesystem

Defence R&D Canada

Canada's Leader in Defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
De science et de technologie pour
la défense et la sécurité nationale

DEFENCE R&D DÉFENSE

www.drdc-rddc.gc.ca