Forensic analysis of SGI IRIX disk volume

R. Carbone
Infosec Analyst & Researcher
EC-Council CHFI
SANS GIAC Certified GCIH & GREM
DRDC – Valcartier Research Centre

Defence Research and Development Canada

Scientific Report
DRDC-RDDC-2016-R127
July 2016

IMPORTANT INFORMATIVE STATEMENTS

The content of this report is not advice and should not be treated as such.

Her Majesty the Queen in right of Canada, as represented by the Minister of National Defence ("Canada"), makes no representations or warranties, express or implied, of any kind whatsoever, and assumes no liability for the accuracy, reliability, completeness, currency or usefulness of any information, product, process or material included in this report. Moreover, nothing in this report should be interpreted as an endorsement of the specific use of any of the tools or techniques examined in it.

Any reliance on, or use of, any information, product, process or material included in this report is at the sole risk of the person so using it or relying on it.

Canada does not assume any liability in respect of any damages or losses arising out of or in connection with the use of, or reliance on, any information, product, process or material included in this report.

Abstract

This report examines the partition structures of SGI IRIX disk and optical media, from the perspective of digital forensics. To date, these structures are poorly documented. Should investigators encounter such systems and media, they may have difficulty obtaining meaningful information and evidence from such devices. This report hopes to provide sufficient information to aid investigators with respect to SGI's proprietary UNIX environment.

Significance to defence and security

Any computer forensic investigator or analyst charged with acquiring or examining evidence from SGI computer media will find this report helpful. In the civilian sector, numerous SGI supercomputers and NUMA systems were sold to academia, R&D agencies, facilities and institutions, and to businesses alike. For government and military, these systems were acquired primarily by organizations and agencies conducting R&D, simulations and numerical calculations, some of which are affiliated to DND and DOD. Law enforcement, whether civilian or military, may be called upon to investigate or acquire evidence from such systems, as many remain in service today. The information in this report will be of use to those acquiring evidence from such systems.

Résumé

Le présent rapport examine les structures de partition des disques et des supports optiques IRIX de SGI dans une perspective d'expertise judiciaire en informatique. À ce jour, ces structures sont peu documentées. Si un enquêteur devait se heurter à des systèmes et des supports de ce type, il pourrait avoir des difficultés à obtenir de l'information et des preuves valables. Le rapport, souhaitons-le, fournit suffisamment d'information pour aider les enquêteurs relativement à l'environnement UNIX propre à SGI.

Importance pour la défense et la sécurité

Tout enquêteur en informatique judiciaire ou analyste chargé d'obtenir ou d'examiner les éléments de preuve d'un support informatique de SGI trouvera ce rapport utile. Dans le domaine civil, de nombreux superordinateurs SGI et systèmes NUMA ont été vendus tant aux universités, aux organismes, aux institutions et aux installations de recherche et développement qu'aux entreprises. Dans le cas des gouvernements et des forces militaires, ces systèmes ont été achetés principalement par des organisations qui mènent de la recherche et du développement, des simulations et des calculs numériques. Certaines de ces organisations sont affiliées au MDN et au Département de la défense des États-Unis. Les forces de l'ordre, civiles ou militaires, pourraient être appelées à enquêter sur ces systèmes ou à en obtenir des éléments de preuve, puisque nombre de ces systèmes demeurent encore en service de nos jours. L'information contenue dans ce rapport pourrait se révéler utile aux personnes qui recherchent des éléments de preuve dans ces systèmes.

Table of contents

List of tables

Acknowledgements

The author would like to thank Mr. Martin Salois, Defence Scientist, for peer reviewing this report.

Disclaimer Policy

It must be understood that this report examines computer forensic technologies, which is not without risk. As such, the reader must be well acquainted with computer and forensic procedure in order to avoid damaging computer systems, both his own and that of others.

The reader must neither construe nor interpret the work described herein by the author as an endorsement of the aforementioned techniques and capacities as suitable for any specific purpose, construed, implied or otherwise. The use of the software discussed herein is neither an endorsement nor a recommendation of said software and tools. Such software and tools should only be used by knowledgeable individuals aware of the inherent risks in using said software and tools.

Furthermore, the author of this report absolves himself in all ways conceivable with respect to how the reader may use, interpret or construe this report. The author assumes absolutely no liability or responsibility, implied or explicit. Moreover, the onus is on the reader to be appropriately equipped and knowledgeable in the application of digital forensics.

Finally, the author and the Government of Canada are henceforth absolved from all wrongdoing, whether intentional, unintentional, construed or misunderstood on the part of the reader. If the reader does not agree to these terms, then his copy of this Scientific Report must be destroyed. Only if the reader agrees to these terms should he continue in reading it beyond this point. It is further assumed by all participants that if the reader has not read said Disclaimer upon reading this report and has acted upon its contents then the reader assumes all responsibility for any repercussions that may result from the information and data contained herein.

Assumptions, Limitations and Exclusions

This report applies to SGI IRIX computer disk and optical media (CD and DVD) as well as forensically acquired images of said media. SGI MIPS-based computer systems ran IRIX, a proprietary UNIX operating system. IRIX, now obsolete, was a very advanced computer operating system that employed a unique partitioning scheme, or volume header, for storing data onto various IRIX-specific media.

Disk partitioning schemes vary greatly by architecture, platform, and operating system. The method for accessing data within an IRIX partition is unlike anything else in the UNIX world of computing, although shares some similarity to other partitioning structures.

IRIX natively supported XFS and EFS. Of these filesystems, EFS was superseded by XFS, which was eventually released by SGI as an open source project.

SGI no longer manufactures its MIPS line of computer systems. However, due to SGI's dominance in the field of NUMA systems and supercomputers, these systems continue to remain relevant today because of the large investment required to procure them; thus, many continue to remain in service. At the same time, IRIX was a particularly well designed UNIX system, and to continues to fair well against other modern UNIX systems.

This analysis will be carried out using a modern Linux system supporting IRIX disk volumes.

This page intentionally left blank.

1 Introduction

1.1 Objective

The objective of this report is to examine the fundamental partitioning structure employed by SGI IRIX for disk and optical media. The underlying structure is the same for all IRIX-encoded media, found across SGI's gamut of workstations, servers, data centers and supercomputers.

There is currently very little publicly available information concerning the overall structure of the SGI disk volume. What is available is in the form of source code, particularly C header files, which are not particularly useful to non-programmers as it pertains to the location and significance of particular byte pattern and signatures.

It is hoped that this report will provide sufficient information to aid forensic investigators charged with examining or acquiring evidence from such systems.

1.2 Background

From IRIX sprang many technologies that are now taken for granted; these include OpenGL, visual supercomputing, VR applications and APIs, and the XFS file system. OpenGL Multipipe [10, 57], OpenGL Performer [56] and VizServer [8, 9, 33, 58] also have had an impact on the industry [55]. At the time when SGI was still manufacturing and supporting IRIX, XFS was considered the computing world's most scalable and likely its fastest filesystem [24–28].

Since XFS was open-sourced in May 2001 [24], Linux has supported not only XFS but also IRIX disk volumes [24, 26, 28–32]. Also since the era of IRIX disk volumes and the DOS MBR, improved partitioning schemes such as GPT [34] have become commonplace to support today's larger hardware storage devices. Due to the prevalence of SGI IRIX systems in the marketplace, disks and media from these systems will continue to be encountered.

Today, XFS is used for an array of applications. Found in the homes of Linux hobbyists, business and government data centers, it is commonly found in high-end storage devices such as network-attached storage (NAS) and storage area network (SAN) systems [35].

Originally, the IRIX disk volume deployed only EFS (Extent File System) [36, 37], the original SGI filesystem. Having been superseded by XFS, EFS is now found only on much older IRIX disks and optical media. In fact, IRIX optical media exclusively employs EFS. Of course, IRIX supports the standard CD and DVD filesystems for maintaining compatibility with non-IRIX systems.

From a digital forensics perspective, the IRIX disk volume structure remains poorly understood. While XFS is well-documented [38], the underlying disk volume structure is not. Unlike other UNIX and Linux filesystems covered in Brian Carrier's seminal *File System Forensic Analysis* [52], XFS was not nor is it supported by Carrier's creation, *The Sleuth Kit*. In fact, with the exception of WinHex Forensics, no other digital forensic analysis tool or framework directly supports XFS filesystem forensics [39].

Thus, not surprisingly, no known framework or tool forensically supports IRIX disk volumes. Immediate sources of information concerning its structure can be found in the C header files of the IRIX operating system, specifically file `/usr/include/sys/dvh.h`. Other sources of information include the source code and manuals of various modern Linux partitioning tools including *sfdisk*, *fdisk*, and others, which support IRIX disk volumes.

1.3 About IRIX

IRIX is similar in functionality to other commercial implementations of UNIX. Although now obsolete, it was at one time a leader in massive Symmetric Multi-Processing (SMP), SSI (Single System Image) and NUMA-based (Non-Uniform Memory Access) computing. Because of the impact SGI and IRIX had in the field of supercomputing, these systems are still in use today [17, 40–47].

Investigators with a solid understanding of Linux or other UNIX operating system will find their way under IRIX in a relatively short period. Although unlikely, it is possible that investigators will encounter Trusted IRIX. It was an IRIX security add-on that provided strict security restrictions from the Orange Book's B1 category [2, 48]. Such systems were typically found in classified and other high security environments.

2 Background information

This section provides an appropriate background to understand the volume structure presented in Section 3.

2.1 Experimentation

All experimentation was conducted using a Linux Fedora 23 x64-based system. Tests were done against actual SGI IRIX system and option disks ranging in size from 18, 36, 72, 147 and 300 GB.

Through simplified reverse engineering, it is possible to identify most IRIX disk volume substructures. With these structures now clearly identified, it is possible for an investigator to readily copy or carve out filesystems or raw data from an IRIX disk or encoded optical media.

To work with IRIX disk volumes under Linux disk-partitioning tools are needed, such as *fdisk*. To acquire forensic copies of IRIX disks and optical media, *dd* or other similar tool can be used. A write-blocker is recommended for disk-based acquisition. Note that IRIX systems used a variety of disk interfaces including SCSI (Small Computer System Interface), fiber channel and ESDI (Enhanced Small Disk Interface)—write blockers exist for these technologies, though some are more difficult and costly to acquire.

To query an IRIX disk volume header, *dvhtool*[1] is recommended. Version 1.01 of the tool was used for this report. Finally, the Linux kernel must have SGI IRIX disk support compiled in; this is the case for Fedora and many other common distributions. XFS is very popular so its support is nearly universal for Linux today; however, support for EFS is not very common. As such, it may be necessary to compile support for it from the kernel's source code.

2.2 Important terminology and other significant information

Many different terms are used throughout the various manuals and documentation, many of which are used in this report. Table 1 contains a list of the most important terms to know.

Table 1: Important terms (source = [1, 52–54]; certain terms amended by author).

Term	Meaning
Bad Block	A defective physical disk block = a bad sector.
Defect List	A list of defective disk blocks for a given disk. Sometimes stored on the disk itself, and sometimes when small enough, within the disk's firmware.

[1] See ftp://ftp.debian.org/debian/pool/main/d/dvhtool.

Disc	An optical disc based media. In the context of this report, it is an IRIX-encoded disc.
Disk	Hard disk drive = HDD.
Disk Block (DB)	This is a physical disk sector = 512 bytes (it seems to always be this size under IRIX).
Disk Header (DH)	This is the disk volume's signature = 55 AA for the MBR.
Disk Volume (DV)	This is the entire disk (or disk).
Disk Volume Header (DVH or VOLHDR)	This includes the first 4,096 blocks of the disk or disc. The first block contains the VTOC (see below) and the second block is empty. Only blocks 3 to 4,096 contain boot-related data files (e.g., sgilabel, SASH) stored using a very simple file storage volume. In short, it is the first 2 MiB of a disk/disc.
Logical Volume (LV)	This is an IRIX-based storage method for allocating multiple disks to create much larger virtual mass-storage devices. Under IRIX, LV is known as XLV.
Lvol	This is a logical volume data or volume set. It is obsolete and no longer supported by IRIX.
MBR	Master Boot Record (PC) = Partition Table = VTOC (see below).
Optical media	CD & DVD. SGI IRIX-encoded media are partitioned with a VTOC (see below), have a DVH and use 512-byte disk blocks. Because of the block size, CD/DVD/Blu-Ray drives must support this block size (not all do). These discs use EFS exclusively. Non-IRIX optical media typically default to the applicable standard (ISO 9660 and its extensions, UDF, etc.).
Option disk	A disk used for data storage, databases or other non-operating system use.
Partition Table	This is the very first disk sector of every disk and disc. It contains the first 512 bytes and defines the partitioned structure of the underlying media. It is equivalent in concept to the VTOC (see below) or MBR.
Raw disk	Primarily used by database systems that directly manage raw disk media.
SASH (Stand Alone SHell)	Non-kernel system shell booted from the DVH used to recover the system. It should be stored within the DVH of every disk, system or option.
Slice	A term commonly used by other UNIX systems to denote a disk partition or volume. Filesystems are written to a slice.

System disk	This is the disk where the operating system is located. A system may have multiple system disks. Booting from a system disk is either specified at the system PROM by the user (for booting alternate disks or discs) or from a pre-existing value.
Usr partition	This is a partition used for user data. This may be /home, /exports, contain a portion of /usr, /var, or other user data.
Volume	A disk partition = Partition = Slice (other UNIX systems). Filesystems are written to a volume.
Volume Header (VH)	VH = DVH.
VTOC	Volume Table of Contents = MBR. The IRIX VTOC only supports 32-bit addressing. IRIX did not support disks larger than 2 TiB. It did, however, support very large logical volumes that could be many petabytes in size, or larger.

The IRIX VTOC neither requires nor supports the use of MBR-style boot code, as is done with the PC MBR and now, the GPT (GUID Partition Table). Booting is done by the system PROM (Programmable Read-Only Memory). Only after the system is initialized does the PROM boot the UNIX kernel file. The kernel file name and location is stored within the VTOC. The PROM needs only the name and location of the disk/disc to boot from. This value may already be stored in the PROM, which is often the default and is used for an unattended system boot. Otherwise, from the PROM's system console, the user can explicitly define which disk/disc is to be booted. That disk/disc is only booted if a UNIX kernel file is defined in its VTOC.

The kernel boot file can be a maximum of 16 characters in length, including the "/" character. It must point to the actual operating system kernel otherwise the disk's operating system will not be bootable.

When the operating system cannot be loaded, perhaps due to a catastrophic crash or other event, SASH can be booted to recover or fix the system.

IRIX provides support for additional filesystems including DOS FAT (File Allocation Table) and Mac HFS (Hierarchal File System) for floppy-only media, and UDF and ISO 9660 for optical media [1, 52–54]. These IRIX filesystems are typically stored on non-IRIX encoded media, that is, media originating from other computer systems.

An IRIX VTOC consists of the very first 2 disk sectors or 1,024 bytes. It is not known if IRIX supports different disk block sizes.

Almost all SCSI disks at this time used 512-byte blocks. Some may have used more, but, for the purposes of this report, it is assumed that all disks encountered on an IRIX system will use this size. It is not known if IRIX can accommodate disks using block sizes other than 512 bytes.

The VOLHDR (Volume Header) file storage volume, sectors 3 to 4,096, can hold a maximum of 15 files, which, together, must not exceed a cumulative size greater than 2 MiB—3 disk blocks or 2,095,616 bytes. Files are aligned on disk sector boundaries.

Bad block management was often done by the SCSI disk controller but could also be initiated from the IRIX *fx* tool.

VOLHDR file *sgilabel* contains information about the underlying disk, specifically the model and make of the disk. It is not known what this information is used for but this is not needed to boot a system. Within the VOLHDR, many other files may exist. These are used to boot the system according to the underlying MIPS (Microprocessor without Interlocked Pipeline Stages) architecture—some only support 32 or 64-bit while others support specific disk interfaces. VOLHDR booting is complex and requires reading the SGI documentation.

The system disk is mounted as "/" or "root" whereas an option disk is either mounted somewhere within the root arborescence or is a raw disk. A raw disk, another type of option disk, requires a VTOC. All hard disks used in a SGI IRIX system must use a VTOC. At a minimum, all IRIX-encoded media must have partitions 8 and 10 defined; otherwise, they will not be useable and may lead to system stability issues. Based on the author's experience, it is possible to crash IRIX using malformed VTOCs. Partition 8 defines the VTOC—that is a partition for storing the partition table which Partition 10 defines the entire physical disk.

Linux-specific partitions and filesystems are not examined in this report, although they can be appended to an IRIX-encoded disk. It is also possible to format an IRIX partition to hold a Linux filesystem.

2.3 IRIX disk management

Under IRIX, the base commands for working with media, modifying and querying IRIX disk volumes are *dvhtool*, *fx, hinv, mkfs*, *prtvtoc*, *xdkm*, and; other commands may be used, depending on media storage types (e.g., external RAID, SAN, which may use third-party tools or software). The system root user or other configured super-user is the only one who can use these commands. These tools are further defined in Table 2.

Table 2: IRIX system commands for working with IRIX-based disks (Source = [1, 53, 54]).

Command	**Capability**
dvhtool	Read and write information to the disk volume; used to put a copy of SASH and other files in volume header
fx	Tool used to initialize disks, format, check for bad blocks, label and relabel disk volumes
hinv	List hardware inventory including disks and tapes
mkfs	Make filesystem command for XFS and EFS
mkswap	Make swap filesystem
prtvtoc	Print information about a disk's volume
xdkm	GUI tool for repartitioning disks

2.4 IRIX disk management under Linux

Under Linux, commands *dvhtool*, *fdisk*, *mkfs*, *mount* and can be used to work disk IRIX disk volumes and filesystems. Specifics are listed in Table 3.

Table 3: Linux system commands for working with IRIX-based disks.

Command	Capability
dvhtool	Read and write information to the volume header; used to put a copy of SASH and other files in volume header. Tool is very similar to IRIX tool.
fdisk	Standard command line disk-partitioning tool for Linux.
mkfs	Make filesystem command for XFS; EFS filesystem creation not supported under Linux.
mount	This command mounts a filesystem. The filesystem must be supported by the kernel (i.e., compiled in) or as a FUSE filesystem.

2.5 IRIX disk and partition types

IRIX supports two basic disk types: system and option disks. A raw disk is an extension of the option disk. A system disk is where the operating system or system configuration information resides. This may also be a dedicated swap disk. Typically, a system will only have one system disk, but additional system disks can exist for booting alternate environments or for recovering the system. Multiple system disks existing on the same system do not typically affect IRIX in any meaningful way, but this may vary by version of IRIX and its underlying system hardware support. [1, 52, 53]

All IRIX-encoded media, including IRIX-specific optical media, must have partitions 8 and 10 defined. Partition 8 is the VTOC itself, where all other partition information is stored. Partition 10 is used to define the actual disk volume size. As such, all system and option disks will have these two partitions in use. The only exception is raw disks, which typically only have Partition 10 defined. [1, 52, 53]

This report will only explore system and option disks. The differences between system and option disks are further explored in Table 4.

Table 4: Differences between System and Option disks (source = [53, 54]).

System disk	Option disk
Booting IRIX	Storing usr data
Storing configurations	Data partition
Has a DV and DVH	Has a DV and DVH (DVH is empty)
Has a copy of SASH and other key files in DVH	Empty DVH

May contain a swap partition	May contain a swap partition
Contains the root filesystem	Contains Partition type 7, others types can be used
Needs Partition type 10	Needs Partition type 10
May contain a usr filesystem	May contain a usr filesystem
The root filesystem is never part of an LV; using system disk for an LV will affect its performance	May be part of an LV
Needs Partition 8	Raw disks do not need Partition 8; standard option disks do

In all, IRIX supports up to a maximum of 16 partitions per disk. Each disk can contain only one DH, DV and DVH. Not all partitions can actually be used to contain data or filesystems.

Table 5 provides a description of the various partition types, 0 to 15. Table 6 provides a description of partition-based filesystem support.

Table 5: Standard partition numbers and descriptions (source = [1, 53]; portions amended by author).

Partition number	**Description**
0	Root partition ("/")
1	Swap partition (raw)
2, 3, 4, 5	Custom partitions—non-IRIX filesystems can go here—can be used as optional root partition
6	Partition for usr filesystem
7	Accessible space (= Disk – DVH – xfslog)
8	Volume header (volhdr)
9	Reserved partition (typically for Bad Block list). It is possible to create a filesystem on this partition such as XVM[2].
10	This is a meta-partition which overlays the entire disk volume (including the DVH). It is sometimes used by databases and video data archiving.
11, 12, 13, 14	Custom partitions—non-IRIX filesystems can go here—may be possible to use as optional root partition (this requires further study with an actual running system)
15	XFS external log (xfslog)

[2] XVM is used in CXFS (Clustered XFS), which is not discussed in this report—only XLV and XFS are.

Table 6: Partition-based filesystem support (source = [1, 53]; portions amended by author).

Partition type	Filesystem type	Partitions that can be this type
EFS	EFS	Standard partitions: 0, 6, 7 Custom partitions: 2, 3, 4, 5, 11, 12, 13, 14, 15
XFS	XFS	Standard partitions: 0, 6, 7 Custom partitions: 2, 3, 4, 5, 11, 12, 13, 14, 15
xfslog	External log for an XFS filesystem must be configured as an XLV log subvolume	Standard partitions: 15 Custom partitions: 0, 2, 3, 4, 5, 6, 7, 11, 12, 13, 14
raw	Swap	1
volhdr	Volume header (DVH)	8
volume	Entire volume (disk)	10
xlv	Part of XLV data or real-time subvolume	0, 1, 2, 3, 4, 5, 6, 7, 11, 12, 13, 14, 15 (partitions are changed to type xlv by XLV commands)
lvol	Part of lv logical volume	Now obsolete. Must be converted to XLV.

The takeaway from Tables 5 and 6 are:

- Partitions 7, 8 and 10 are reserved.
- The location of Partition 8 is non-negotiable.
- The size of Partition 10 is based entirely on the number of physical disk sectors (or blocks) of the underlying disk.
- Partition 7 is based on the number of available disk blocks.
- The root filesystem (XFS or EFS) can be on any partition number except for 1, 8, 9, 10 and 15. The same applies for usr filesystems.
- Swap partitions must be set to partition number 1.
- Lvol volumes are no longer supported.
- XLV volumes can use any partition number except for 8, 9 and 10.
- XFS filesystems can use external an xfslog partition but which must be set to XLV log subvolume.

Fiddling with partitions types, numbers and locations can give system administrators and other root users the ability to hide data in many places. It is possible to hide data within an IRIX disk or disc by fiddling with the locations of partitions, the VOLHDR, or other reserved partition space. Such actions can only be done by root. However, changes made to the VTOC cannot be made lightly as they can significantly affect system stability or cause existing data to be overwritten.

Data carving can be used to recover information and data including lost or hidden partitions and filesystems. Data recovery tools based on file signature detection can recover known file types,

intact and damaged alike; however, they will not be able to recover a deleted partition or reformatted filesystem from an IRIX system. Fortunately, for manual data recovery, and based on the existing values of other partitions, it may be possible to accurately estimate the location of a lost partition.

Under IRIX, disk partitioning is done using *fx*, but this can also be done before the operating system is booted by using a DVH-stored version of *fx*, or from an IRIX installation disc.

Linux can make use of most IRIX partitions without interfering with IRIX. Specifics are not available at this time.

2.6 IRIX imposed limits for partitions

Circa 2005/06, SGI systems were using almost exclusively SCSI (50, 68 and 80 pins) or fiber channel disks. Typically, a system disk will be an internal disk, but it does not have to be—it could be an external disk connected via fiber channel or SCSI. Although some systems may use fiber channel interfaces, the underlying disks are essentially the same as their SCSI counterparts. At this time, older SCSI disks had given way to Ultra SCSI, Ultra Wide SCSI and other variations.

In 2006, SCSI disk sizes were between 73 and 300 GB and UltraSCSI-320, 300 GB, 15K RPM disks retailed for approximately USD700-1000. In 2009, SGI filed for bankruptcy protection [55].

Although IRIX handled petabyte-sized filesystems spanning across numerous disk media, individual disk size support was limited at 2 TiB, the limit of 32-bit disk partitioning technology.

2.7 About IRIX optical media

IRIX-specific optical media such as SGI installation and software discs must contain a disk volume, use EFS and have a disc block size of 512 bytes. Standard non-IRIX optical media are supported; these include UDF, ISO 9660, High Sierra, Photo CD and Music CD formats. These can use standard optical device-based block sizes [1, 53, 54].

To acquire evidence from such a disc requires only a low-level acquisition tool such as *dd* and an optical device supporting 512-byte disk blocks. Once the disc's VTOC has been parsed, manually or using *dvhtool*, its EFS filesystem(s) can be mounted, assuming support is available in the kernel. The same overall rules for disk partitioning apply to optical discs.

2.8 Issues with *dvhtool*

Dvhtool, as practical as it is, does not handle sizes of more than about 1 TiB. Beyond that, it recognizes them as negatively sized partitions. Obviously, a disk partition cannot be negative. *Dvhtool* is likely the best and most direct tool for interacting with the SGI DVH and its files—for this reason, its continued use is suggested.

Linux and other operating systems (e.g., BSD) using *fdisk* or other equivalent disk partitioning supporting IRIX VTOCs should be used to identify the sizes of partitions greater than

approximately 1 TiB. Since IRIX-specific hard disk drives were usually smaller than this, this situation will rarely be seen; however, this particular case in shown in the example of Section 3.

2.9 Disk imaging, data/evidence acquisition and XFS log journals

Write blockers are readily available for offline and powered off SCSI-based disks; adapters are available from various manufacturers to support all the numerous incarnations of SCSI and its interfaces. For fiber channel based disks, though harder to find, write blockers also exist. For optical media, they can be readily imaged from a system supporting 512-byte sectors. For disk-based media various Linux or Windows disk-imaging software can be used; for optical media, it is strongly suggested to use a Linux or UNIX version of *dd* or similar software (e.g., *dcfldd*, *dc3dd*.)

Problems with acquisition and future forensic analysis largely stem from acquisition against a live IRIX system. Live systems cannot have their disks disconnected, attached to a write blocker and then reattached—this simply will not do. Although IRIX includes the *dd* command, which the root user can use to image online disks/discs, disk acquisition against mounted filesystems is filled with problems. IRIX XFS is not the same creature that modern Linux XFS is; there are many features present in the modern implementation that were not there then, such as *xfs_freeze*, a command that freezes XFS I/O. This command is not available under IRIX, meaning that filesystem I/O must be managed through a volume manager (e.g., XLV or XVM) and not the operating system. For an optical media, this is not an issue as it is read-only.

To successfully acquire an XFS filesystem, it is strongly suggested that it be unmounted. Sometimes, it is simply not possible due to system or usability constraints. To minimize issues during acquisition, ensure that the filesystem is quiescent and that filesystem memory buffers have been synced to disk (e.g., using the *sync* command). Disk synchronizing under IRIX is not straightforward; under Linux, the *sync* command returns to the prompt when all buffers have been flushed. IRIX, on the other hand, does not [59].

Thus, unfortunately, acquisition against a mounted XFS filesystem is a best-effort initiative. While data recovery software such as carvers and signature detection utilities and programs can be ran against the image from a mounted filesystem, mounting may altogether be something else. The *mount* command will likely refuse or fail. This is because the XFS filesystem is in a dirty state and needs fixing (e.g., *xfs_check* and *xfs_repair*). The inconsistency is due to the ongoing changes that were made to the filesystem, changes coming from data being written to disk, unwritten buffers in memory and discrepancies between what is in a disk image, as it is being generated from the live filesystem, to what is written later on therein.

The problem with fixing is that it may introduce many filesystem changes. Unfortunately, there are no hard facts concerning this issue so the decision to acquire a live filesystem ultimately lies with the investigator.

Although XFS is a journaling filesystem, IRIX XFS requires that partition-based XFS filesystems use internal journals, meaning that there is no need to identify external log devices. Acquiring an XFS filesystem without its journal has the potential to lead to widespread filesystem corruption, even though XFS is a particularly robust filesystem. Fortunately, this is not the case since the journal will be in the filesystem.

3 IRIX disk volume structure

3.1 VTOC on-disk layout

The best way to describe abstract concepts is often with an example. The example shown in Table 7 is the VTOC from an SGI IRIX-encoded disk. The disk image, created from a sparse file, has been allocated 4,294,967,296 disk sectors or exactly 2 TiB. The image is fully populated with DVH files and 16 partitions. The VTOC fields are examined in the following subsections.

Table 7: SGI IRIX VTOC—a populated example.

OFFSET	00 01 02 03	04 05 06 07	08 09 0A 0B	0C 0D 0E 0F		DISK SECTOR 0
00 (1)	0B E5 A9 41	00 0C 00 00	2F 75 6E 69	78 2E 6F 6C	(16)	.å©A..../unix.ol
01 (17)	64 2E 62 6F	6F 74 31 31	00 00 00 00	14 55 00 00	(32)	d.boot11....z-..
02 (33)	00 FF 00 00	00 00 00 3F	02 00 00 01	00 00 00 34	(48)	.......?.......4
03 (49)	00 00 00 00	00 00 00 01	00 00 00 00	00 00 00 00	(64)	....
04 (65)	00 00 00 00	08 8B B9 98	73 67 69 6C	61 62 65 6C	(80)	........sgilabel
05 (81)	00 00 00 04	00 00 02 00	73 79 6D 6D	6F 6E 00 00	(96)	........symmon..
06 (97)	00 00 00 05	00 17 CC 00	73 61 73 68	00 00 00 00	(112)	........sash....
07 (113)	00 00 0B EB	00 00 08 00	74 65 73 74	31 00 00 00	(128)	........test1...
08 (129)	00 00 0D F5	00 00 08 00	74 65 73 74	32 00 00 00	(144)	........test2...
09 (145)	00 00 0D F9	00 00 08 00	74 65 73 74	33 00 00 00	(160)	........test3...
0A (161)	00 00 0D FD	00 00 08 00	74 65 73 74	34 00 00 00	(176)	........test4...
0B (177)	00 00 0E 01	00 00 08 00	74 65 73 74	35 00 00 00	(192)	........test5...
0C (193)	00 00 0E 05	00 00 08 00	74 65 73 74	36 00 00 00	(208)	........test6...
0D (209)	00 00 0E 09	00 00 08 00	74 65 73 74	37 30 00 00	(224)	........test7...
0E (225)	00 00 0E 0D	00 00 08 00	74 65 73 74	38 00 00 00	(240)	........test8...
0F (241)	00 00 0E 11	00 00 08 00	74 65 73 74	39 00 00 00	(256)	........test9...
10 (257)	00 00 0E 15	00 00 08 00	74 65 73 74	31 30 00 00	(272)	........test10..
11 (273)	00 00 0E 19	00 00 08 00	74 65 73 74	31 31 31 00	(288)	........test111.
12 (289)	00 00 0E 1D	00 00 08 00	74 65 73 74	31 31 00 00	(304)	........test11..
13 (305)	00 00 0E 21	00 00 08 00	19 00 00 01	00 00 3E C1	(320)	..............>.
14 (321)	00 00 00 0A	0C 80 00 01	19 00 68 7E	00 00 00 0A	(336)	..........h~....
15 (337)	0C 80 00 01	25 80 9C BD	00 00 00 0A	0C 80 00 01	(352)	....%...........
16 (353)	32 00 D0 FC	00 00 00 0A	0F A0 00 01	3E 81 05 3B	(368)	2...........>..;
17 (369)	00 00 00 0A	0F A0 00 01	4E 21 17 79	00 00 00 0A	(384)	........N!.y....
18 (385)	0C 80 00 01	5D C1 29 B7	00 00 00 0A	0C 80 00 01	(400)	....].).........
19 (401)	6A 41 5D F6	00 00 00 0A	00 00 10 00	00 00 00 00	(416)	jA].............
1a (417)	00 00 00 00	0C 80 00 01	76 C1 92 35	00 00 00 0A	(432)	........v..5....
1b (433)	FF FF EA 15	00 00 00 00	00 00 00 06	12 C0 00 01	(448)	....
1c (449)	83 41 C6 74	00 00 00 0A	12 C0 00 01	96 01 F5 72	(464)	.A.t...........r
1d (465)	00 00 00 0A	12 C0 00 01	A8 C2 24 70	00 00 00 0A	(480)	..........$p....
1e (481)	12 C0 00 01	BB 82 53 6E	00 00 00 0A	31 BD 67 A9	(496)	......Sn....1.g.
1f (497)	CE 42 82 6C	00 00 00 0A	B2 6D 54 84	00 00 00 00	(512)	.B.l.....mT.....

3.2 VTOC field-based layout and description

Extracting the VTOC from Table 7, it is possible to discern its true structure, as determined through reverse engineering via VTOC modification and comparison. Table 8 fully describes this structure.

Only 37 bytes or 7.2% of the VTOC could not be identified. A full understanding would require analyzing the source code from various IRIX-supporting disk-partitioning tools, a larger undertaking than what could be done in this report. Nevertheless, multiple experiments have shown that these unknown byte values do not have any effect on the actual partition locations or sizes.

Table 8: Byte offset-based layout and description for example IRIX VTOC.

Byte offset	Description
1 - 4	DISK SIGNATURE / THIS IS THE SAME FOR EVERY SGI DISK
5	UNKNOWN
6	BOOTABLE PARTITION ENCODED IN HEX
7 - 8	UNKNOWN
9 - 24	BOOTABLE KERNEL NAME
25 - 56	UNKNOWN
57 - 68	ALWAYS APPEARS TO BE EMPTY
69 - 72	SECTOR SIZE OF PARTITION 10
73 - 80	FILE NAME OF VH 1
81 - 84	OFFSET IN SECTORS FROM START OF DISK FOR VH FILE1
85 - 88	SIZE OF VH FILE1
89 - 96	FILE NAME OF VH 2
97 - 100	OFFSET IN SECTORS FROM START OF DISK FOR VH FILE2
101 - 104	SIZE OF VH FILE2
105 - 112	FILE NAME OF VH 3
113 - 116	OFFSET IN SECTORS FROM START OF DISK FOR VH FILE3
117 - 120	SIZE OF VH FILE3
121 - 128	FILE NAME OF VH 4
129 - 132	OFFSET IN SECTORS FROM START OF DISK FOR VH FILE4
133 - 136	SIZE OF VH FILE4
137 - 144	FILE NAME OF VH 5
145 - 148	OFFSET IN SECTORS FROM START OF DISK FOR VH FILE5
149 - 152	SIZE OF VH FILE5
153 - 160	FILE NAME OF VH 6

161 - 164	OFFSET IN SECTORS FROM START OF DISK FOR VH FILE6
165 - 168	SIZE OF VH FILE6
169 - 176	FILE NAME OF VH 7
177 - 180	OFFSET IN SECTORS FROM START OF DISK FOR VH FILE7
181 - 184	SIZE OF VH FILE7
185 - 192	FILE NAME OF VH 8
193 - 196	OFFSET IN SECTORS FROM START OF DISK FOR VH FILE8
197 - 200	SIZE OF VH FILE8
201 - 208	FILE NAME OF VH 9
209 - 212	OFFSET IN SECTORS FROM START OF DISK FOR VH FILE9
213 - 216	SIZE OF VH FILE9
217 - 224	FILE NAME OF VH 10
225 - 228	OFFSET IN SECTORS FROM START OF DISK FOR VH FILE10
229 - 232	SIZE OF VH FILE10
233 - 240	FILE NAME OF VH 11
241 - 244	OFFSET IN SECTORS FROM START OF DISK FOR VH FILE11
245 - 248	SIZE OF VH FILE11
249 - 256	FILE NAME OF VH 12
257 - 260	OFFSET IN SECTORS FROM START OF DISK FOR VH FILE12
261 - 264	SIZE OF VH FILE12
265 - 272	FILE NAME OF VH 13
273 - 276	OFFSET IN SECTORS FROM START OF DISK FOR VH FILE13
277 - 280	SIZE OF VH FILE13
287 - 288	FILE NAME OF VH 14
289 - 292	OFFSET IN SECTORS FROM START OF DISK FOR VH FILE14
293 - 296	SIZE OF VH FILE14
297 - 304	FILE NAME OF VH 15
305 - 308	OFFSET IN SECTORS FROM START OF DISK FOR VH FILE15
309 - 312	SIZE OF VH FILE15
313 - 316	SIZE IN DISK BLOCKS FOR PARTITION 0
317 - 320	STARTING DISK BLOCK FOR PARTITION 0
321 - 324	PARTITION TYPE/CODE FOR PARTITION 0
325 - 328	SIZE IN DISK BLOCKS FOR PARTITION 1
329 - 332	STARTING DISK BLOCK FOR PARTITION 1
333 - 336	PARTITION TYPE/CODE FOR PARTITION 1

337 - 340	SIZE IN DISK BLOCKS FOR PARTITION 2
341 - 344	STARTING DISK BLOCK FOR PARTITION 2
345 - 348	PARTITION TYPE/CODE FOR PARTITION 2
349 - 352	SIZE IN DISK BLOCKS FOR PARTITION 3
353 - 356	STARTING DISK BLOCK FOR PARTITION 3
357 - 360	PARTITION TYPE/CODE FOR PARTITION 3
361 - 364	SIZE IN DISK BLOCKS FOR PARTITION 4
365 - 368	STARTING DISK BLOCK FOR PARTITION 4
369 - 372	PARTITION TYPE/CODE FOR PARTITION 4
373 - 376	SIZE IN DISK BLOCKS FOR PARTITION 5
377 - 380	STARTING DISK BLOCK FOR PARTITION 5
381 - 384	PARTITION TYPE/CODE FOR PARTITION 5
385 - 388	SIZE IN DISK BLOCKS FOR PARTITION 6
389 - 392	STARTING DISK BLOCK FOR PARTITION 6
393 - 396	PARTITION TYPE/CODE FOR PARTITION 6
397 - 400	SIZE IN DISK BLOCKS FOR PARTITION 7
401 - 404	STARTING DISK BLOCK FOR PARTITION 7
405 - 408	PARTITION TYPE/CODE FOR PARTITION 7
409 - 412	SIZE IN DISK BLOCKS FOR PARTITION 8
413 - 416	STARTING DISK BLOCK FOR PARTITION 8
417 - 420	PARTITION TYPE/CODE FOR PARTITION 8
421 - 424	SIZE IN DISK BLOCKS FOR PARTITION 9
425 - 428	STARTING DISK BLOCK FOR PARTITION 9
429 - 432	PARTITION TYPE/CODE FOR PARTITION 9
433 - 436	SIZE IN DISK BLOCKS FOR PARTITION 10
437 - 440	STARTING DISK BLOCK FOR PARTITION 10
441 - 444	PARTITION TYPE/CODE FOR PARTITION 10
445 - 448	SIZE IN DISK BLOCKS FOR PARTITION 11
449 - 452	STARTING DISK BLOCK FOR PARTITION 11
453 - 456	PARTITION TYPE/CODE FOR PARTITION 11
457 - 460	SIZE IN DISK BLOCKS FOR PARTITION 12
461 - 464	STARTING DISK BLOCK FOR PARTITION 12
465 - 468	PARTITION TYPE/CODE FOR PARTITION 12
469 - 472	SIZE IN DISK BLOCKS FOR PARTITION 13
473 - 476	STARTING DISK BLOCK FOR PARTITION 13

477 - 480	PARTITION TYPE/CODE FOR PARTITION 13
481 - 484	SIZE IN DISK BLOCKS FOR PARTITION 14
485 - 488	STARTING DISK BLOCK FOR PARTITION 14
489 - 492	PARTITION TYPE/CODE FOR PARTITION 14
493 - 496	SIZE IN DISK BLOCKS FOR PARTITION 15
497 - 500	STARTING DISK BLOCK FOR PARTITION 15
501 - 504	PARTITION TYPE/CODE FOR PARTITION 15
505 - 508	UNKNOWN
509 - 512	ALWAYS APPEARS TO BE EMPTY

3.3 Corresponding VTOC values

The corresponding values to the various fields identified in Table 8 are shown in Table 9.

In this table, incorrect values obtained from *dvhtool* for the partition sizes are shown; incorrect values are highlighted in blue. The red text indicates the corresponding numerical values for the partition sizes.

The calculation issues with *dvhtool* begin when reaching Partition 10 (the meta-partition), which spans the entire disk. Corrections to *dvhtool*'s incorrect calculation are provided in Section 3.4.

Table 9: Corresponding values for example IRIX VTOC (partition size and starting location obtained using dvhtool).

Byte offset	Example value
1 - 4	0x0B E5 A9 41 = .å©A
6	0x0C = 12
9 - 24	0x2F 75 6E 69 78 2E 6F 6C 64 2E 62 6F 6F 74 31 31 = /unix.old.boot11
57 - 68	0x00 00 00 00 00 00 00 00 00 00 00 00 = 0
69 - 72	0x08 8B B9 98 = 143374744
73 - 80	0x73 67 69 6C 61 62 65 6C = sgilabel
81 - 84	0x00 00 00 04 = 4
85 - 88	0x00 00 02 00 = 512
89 - 96	0x73 79 6D 6D 6F 6E 00 00 = symmon
97 - 100	0x00 00 00 05 = 5
101 - 104	0x00 17 CC 00 = 1559552
105 - 112	0x73 61 73 68 00 00 00 00 = sash
113 - 116	0x00 00 0B EB = 3051
117 - 120	0x00 00 08 00 = 2048
121 - 128	0x74 65 73 74 31 00 00 00 = test1

129 - 132	0x00 00 0D F5 = 3573
133 - 136	0x00 00 08 00 = 2048
137 - 144	0x74 65 73 74 32 00 00 00 = test2
145 - 148	0x00 00 0D F9 = 3577
149 - 152	0x00 00 08 00 = 2048
153 - 160	0x74 65 73 74 33 00 00 00 = test3
161 - 164	0x00 00 0D FD = 3581
165 - 168	0x00 00 08 00 = 2048
169 - 176	0x74 65 73 74 34 00 00 00 = test4
177 - 180	0x00 00 0E 01 = 3585
181 - 184	0x00 00 08 00 = 2048
185 - 192	0x74 65 73 74 35 00 00 00 = test5
193 - 196	0x00 00 0E 05 = 3589
197 - 200	0x00 00 08 00 = 2048
201 - 208	0x74 65 73 74 36 00 00 00 = test6
209 - 212	0x00 00 0E 09 = 3593
213 - 216	0x00 00 08 00 = 2048
217 - 224	0x74 65 73 74 37 00 00 00 = test7
225 - 228	0x00 00 0E 0D = 3597
229 - 232	0x00 00 08 00 = 2048
233 - 240	0x74 65 73 74 38 00 00 00 = test8
241 - 244	0x00 00 0E 11 = 3601
245 - 248	0x00 00 08 00 = 2048
249 - 256	0x74 65 73 74 39 00 00 00 = test9
257 - 260	0x00 00 0E 15 = 3605
261 - 264	0x00 00 08 00 = 2048
265 - 272	0x74 65 73 74 31 30 00 00 = test10
273 - 276	0x00 00 0E 19 = 3609
277 - 280	0x00 00 08 00 = 2048
287 - 288	0x74 65 73 74 31 31 31 00 = test111
289 - 292	0x00 00 0E 1D = 3613
293 - 296	0x00 00 08 00 = 2048
297 - 304	0x74 65 73 74 31 31 00 00 = test11
305 - 308	0x00 00 0E 21 = 3617
309 - 312	0x00 00 08 00 = 2048
	Partition 0
313 - 316	0x19 00 00 01 = 419 430 401
	SIZE (IN BYTES) = 214 748 364 800
317 - 320	0x00 00 3E C1 = 16065
	STARTING POSITION (IN BYTES) = 8225280

321 - 324	0x00 00 00 0A = TYPE: XFS VOLUME
	Partition 1
325 - 328	0x0C 80 00 01 = 209 715 201
	SIZE (IN BYTES) = 107 374 182 912
329 - 332	0x19 00 68 7E = 419 457 150
	STARTING POSITION (IN BYTES) = 214 762 060 800
333 - 336	0x00 00 00 0A = TYPE: XFS VOLUME
	Partition 2
337 - 340	0x0C 80 00 01 = 209 715 201
	SIZE (IN BYTES) = 107 374 182 912
341 - 344	0x25 80 9C BD = 629 185 725
	STARTING POSITION (IN BYTES) = 322 143 091 200
345 - 348	0x00 00 00 0A = TYPE: XFS VOLUME
	Partition 3
349 - 352	0x0C 80 00 01 = 209 715 201
	SIZE (IN BYTES) = 107 374 182 912
353 - 356	0x32 00 D0 FC = 838 914 300
	STARTING POSITION (IN BYTES) = 429 524 121 600
357 - 360	0x00 00 00 0A = TYPE: XFS VOLUME
	Partition 4
361 - 364	0x0F A0 00 01 = 262 144 001
	SIZE (IN BYTES) = 134 217 728 512
365 - 368	0x3E 81 05 3B = 1 048 642 875
	STARTING POSITION (IN BYTES) = 536 905 152 000
369 - 372	0x00 00 00 0A = TYPE: XFS VOLUME
	Partition 5
373 - 376	0x0F A0 00 01 = 262 144 001
	SIZE (IN BYTES) = 134 217 728 512
377 - 380	0x4E 21 17 79 = 1 310 791 545
	STARTING POSITION (IN BYTES) = 671 125 271 040
381 - 384	0x00 00 00 0A = TYPE: XFS VOLUME
	Partition 6
385 - 388	0x0C 80 00 01 = 209 715 201
	SIZE (IN BYTES) = 107 374 182 912
389 - 392	0x5D C1 29 B7 = 1 572 940 215
	STARTING POSITION (IN BYTES) = 805 345 390 080

393 - 396	0x00 00 00 0A = TYPE: XFS VOLUME
	Partition 7
397 - 400	0x0C 80 00 01 = 209 715 201
	SIZE (IN BYTES) = 107 374 182 912
401 - 404	0x6A 41 5D 01 = 1 782 668 790
	STARTING POSITION (IN BYTES) = 912 726 420 480
405 - 408	0x00 00 00 0A = TYPE: XFS VOLUME
	Partition 8
409 - 412	0x00 00 10 00 = 4 096
	SIZE (IN BYTES) = 2 097 152
413 - 416	0x00 00 00 00 = 0
	STARTING POSITION (IN BYTES) = 0
417 - 420	0x00 00 00 00 = TYPE: VOLUME HEADER
	Partition 9
421 - 424	0x0C 80 00 01 = 209 715 201
	SIZE (IN BYTES) = 107 374 182 912
425 - 428	0x76 C1 92 35 = 1 992 397 365
	STARTING POSITION (IN BYTES) = 1 020 107 450 880
429 - 432	0x00 00 00 0A = TYPE: XFS VOLUME
	Partition 10
433 - 436	0xFF FF EA 15 = -5 611
	SIZE (IN BYTES) = -2 872 832
437 - 440	0x00 00 00 00 = 0
	STARTING POSITION (IN BYTES) = 0
441 - 444	0x00 00 00 06 = TYPE: VOLUME
	Partition 11
445 - 448	0x12 C0 00 01 = 314 572 801
	SIZE (IN BYTES) = 161 061 274 112
449 - 452	0x83 41 c6 74 = -2 092 841 356
	STARTING POSITION (IN BYTES) = -1 071 534 774 272
453 - 456	0x00 00 00 0A = TYPE: XFS VOLUME
	Partition 12
457 - 460	0x12 C0 00 01 = 314 572 801
	SIZE (IN BYTES) = 161 061 274 112
461 - 464	0x96 01 F5 72 = -1 778 256 526
	STARTING POSITION (IN BYTES) = -910 467 341 312
465 - 468	0x00 00 00 0A = TYPE: XFS VOLUME
	Partition 13
469 - 472	0x12 C0 00 01 = 314 572 801
	SIZE (IN BYTES) = 161 061 274 112

473 - 476	0xA8 C2 24 70 = -1 463 671 696
	STARTING POSITION (IN BYTES) = -749 399 908 352
477 - 480	0x00 00 00 0A = TYPE: XFS VOLUME
	Partition 14
481 - 484	0X12 C0 00 01 = 314 572 801
	SIZE (IN BYTES) = 161 061 274 112
485 - 488	0xB8 82 53 6E = -1 149 086 866
	STARTING POSITION (IN BYTES) = -588 332 475 392
489 - 492	0x00 00 00 0A = TYPE: XFS VOLUME
	Partition 15
493 - 496	0x31 BD 67 A9 = 834 496 425
	SIZE (IN BYTES) = 427 262 169 600
497 - 500	0xCE 42 82 6C = -834 502 036
	STARTING POSITION (IN BYTES) = -427 265 042 432
501 - 504	0x00 00 00 0A = TYPE: XFS VOLUME

3.4 Identifying true partition sizes and free space

Based only on the hexadecimal values from the VTOC for the various partition fields, calculating the true partition size is straightforward, as shown in Table 10.

Table 10: Precise starting and ending sectors yielding true partition sizes.

Partition	Starting Sector	Ending Sector	Size in Sectors	Size in bytes	Size in GiB
0	16065	419446465	419430401	214,748,365,312	200.000
1	419457150	629172350	209715201	107,374,182,912	100.000
2	629185725	838900925	209715201	107,374,182,912	100.000
3	838914300	1048629500	209715201	107,374,182,912	100.000
4	1048642875	1310786875	262144001	134,217,728,512	125.000
5	1310791545	1572935545	262144001	134,217,728,512	125.000
6	1572940215	1782655415	209715201	107,374,182,912	100.000
7	1782668790	1992383990	209715201	107,374,182,912	100.000
8	0	4095	4096	2,097,152	0.002
9	1992397365	2202112565	209715201	107,374,182,912	100.000
10	0	4294961684	4294961685	2,199,020,382,720	2047.997
11	2202125940	2516698740	314572801	161,061,274,112	150.000
12	2516710770	2831283570	314572801	161,061,274,112	150.000
13	2831295600	3145868400	314572801	161,061,274,112	150.000
14	3145880430	3460453230	314572801	161,061,274,112	150.000
15	3460465260	4294961684	834496425	427,262,169,600	397.919

Between the ending and starting sectors for most of the partitions, there is free space. This free space makes an ideal location for hiding data. Although data can in reality be hidden anywhere within an existing filesystem or partition, this is not always a good idea, especially if RAID or disk-spanning mechanisms are in place as overwriting even small portions of allocated sectors could result in data corruption or system instability. To hide data in this fashion requires administrative or root privileges; thus, it is most likely that anyone doing so is the system administrator. Of course, the system could have been compromised. Table 11 provides addresses and size where data would most likely be hidden.

The values for Table 11 were directly calculated from Table 10, and the value 1 (one) was added or subtracted to that value, accordingly.

As a final note, it is not suggested to hide data in the DVH as it may accidentally be used by the PROM, possibly resulting in PROM corruption or an inability to boot.

Table 11: Available free space between partitions.

Between	From Sector	To Sector	Size in Sectors	Size in Bytes
8-0	4096	16064	11968	6,127,616
0-1	419446466	419457149	10683	5,469,696
1-2	629172351	629185724	13373	6,846,976
2-3	838900926	838914299	13373	6,846,976
3-4	1048629501	1048642874	13373	6,846,976
4-5	1310786876	1310791544	4668	2,390,016
5-6	1572935546	1572940214	4668	2,390,016
6-7	1782655416	1782668789	13373	6,846,976
7-9	1992383991	1992397364	13373	6,846,976
9-11	2202112566	2202125939	13373	6,846,976
11-12	2516698741	2516710769	12028	6,158,336
12-13	2831283571	2831295599	12028	6,158,336
13-14	3145868401	3145880429	12028	6,158,336
14-15	3460453231	3460465259	12028	6,158,336
15-end of disk	4294961685	4294967296	5611	2,872,832

4 Discussion

By today's standard, IRIX is old and obsolete, but it was the pinnacle of technology in the UNIX world at the time. Its disk-partitioning scheme is also a reflection of its advanced system engineering.

Reverse engineering the data in the SGI IRX VTOC was straightforward. All that was required was an IRIX-aware Linux system, a compiled version of *dvhtool* and fdisk and many SGI disk volumes to work with. By manipulating the various partition-based fields, it was possible to isolate and determine the functionality of 92.8% of the VTOC. Although certain portions of the unknown VTOC code could not be determined, they varied neither by disk nor by tool, leaving the strong impression that they are somewhat fixed values or were reserved for future use.

While examining and parsing the source code from various SGI partition-aware tools would have permitted the complete understanding of the SGI IRIX VTOC structure, reverse engineering the data presented an alternate and straightforward mechanism to thoroughly understand it.

With this knowledge in hand, investigators can hopefully tackle an SGI IRIX-encoded media and bit-copy, recover or carve out the desired data or evidence.

References

[1] SGI. IRIX Admin: Disks and Filesystems. Technical manual. Document No. 007-2825-012. SGI. June 2003.

[2] SGI. Trusted IRIXTM/CMW Security Administration Guide. Technical manual. Document No. 007-3299-009. SGI. November 2003.

[8] SGI. SGI OpenGL Vizserver User's Guide. Technical manual. Document No. 007-4245-015. SGI. July 2005. Last Accessed: June 2016. https://research.iat.sfu.ca/wiki/images/e/e0/VizServer_UserGuide_007-4245-015.pdf.

[9] SGI. SGI OpenGL Vizserver Administrator's Guide. Technical manual. Document No. 007-4481-011. SGI. July 2005. Last Accessed: June 2016. https://research.iat.sfu.ca/wiki/images/f/f0/VizServer_AdminGuide_007-4481-011.pdf.

[10] SGI. SGI OpenGL Multipipe User's Guide. Technical manual. Document No. 007-4318-012. SGI. December 2003. Last Accessed: June 2016. http://manx.classiccmp.org/mirror/techpubs.sgi.com/library/manuals/4000/007-4318-012/pdf/007-4318-012.pdf.

[17] PR Newswire. SGI Origin 3800 Powers World's Largest Production Single-System-Image Supercomputer for Commercial Grid Applications. Press communiqué. PR Newswire. January 2002. Last Accessed: June 2016. http://www.prnewswire.com/news-releases/sgi-origin-3800-powers-worlds-largest-production-single-system-image-supercomputer-for-commercial-grid-applications-75603357.html.

[24] Wikipedia, The Free Encyclopedia. XFS. June 2016. Last Accessed: June 2016. https://en.wikipedia.org/wiki/XFS.

[25] Sweeney, Adam. Scalability in the XFS File System. Technical paper. USENIX, Proceedings of the USENIX 1996 Annual Technical Conference, San Diego, California, 1996. 1996. Last Accessed: June 2016. http://www.scs.stanford.edu/nyu/03sp/sched/sgixfs.pdf.

[26] Hellwig, Christoph. XFS: the big storage file system for Linux. Article. ;LOGIN, Vol. 34, No. 5. October 2019. Last Accessed: June 2016. https://www.usenix.org/system/files/login/articles/140-hellwig.pdf.

[27] Edge, Jake. XFS: There and back … and there again? Article. April 2015. LWN.net. Last Accessed: June 2016. https://lwn.net/Articles/638546/.

[28] SGI. Open Source XFS for Linux. Datasheet. SGI. 2006. Last Accessed: June 2016. http://oss.sgi.com/projects/xfs/datasheet.pdf.

[29] Mostek, Jim; Earl, William; Koren, Dan; Cattelan, Russell; Preslan, Kenneth and O'Keefe, Matthew. Porting the SGI File System to Linux. Technical paper. SGI and Sistina Software, Inc. Unknown date. Last Accessed: June 2016. http://oss.sgi.com/projects/xfs/papers/als/als.pdf.

[30] Total Knowledge. SGI O2 BootCD HOW-TO. How-to. Total Knowledge. 2013. Last Accessed: June 2016. http://www.total-knowledge.com/progs/mips/SGI-BootCD-HOWTO.html.

[31] Chantrell, Nathan. SGI Indy and Debian Linux. How-to. Nathan.chantrell.net. Unknown date. Last Accessed: June 2016. https://nathan.chantrell.net/linux/sgi-indy-and-debian-linux/.

[32] Günther, Guido. Linux on SGI MIPS Hard Disk Boot µ-Howto. How-to. Sigxcpu.org. March 2003. Last Accessed: June 2016. http://honk.sigxcpu.org/linux-mips/indy-boot/indy-hd-boot-micro-howto.html.

[33] SGI. SGI VizServer Systems with NICE Software for Remote Visualization Access via Private Clouds and Data Centers. Whitepaper. SGI. September 2013. Last Accessed: June 2016. https://www.sgi.com/pdfs/4429.pdf.

[34] Wikipedia, The Free Encyclopedia. GUID Partition Table. June 2016. Last Accessed: June 2016. https://en.wikipedia.org/wiki/GUID_Partition_Table.

[35] XFS.org. XFS Companies. Online information. XFS.org. February 2012. Last Accessed: June 2016. http://xfs.org/index.php/XFS_Companies.

[36] Hinner, Martin. Filesystems HOWTO. How-to. Version 0.8. The Linux Documentation Project. January 2007. Last Accessed: June 2016. http://www.tldp.org/HOWTO/Filesystems-HOWTO-9.html.

[37] Unknown author. EFS for Linux. Informational web page. Aeschi.ch.eu.org. Unknown date. Last Accessed: July 2016. http://aeschi.ch.eu.org/efs/.

[38] SGI. XFS Filesystem Structure. Technical document. Second Edition, First Revision. SGI. 2006. Last Accessed: June 2016. http://oss.sgi.com/projects/xfs/papers/xfs_filesystem_structure.pdf.

[39] X-Ways. X-Ways Forensics: Integrated Computer Forensics Software. Software product web page. X-Ways. 2016. Last Accessed: June 2016. http://www.x-ways.net/forensics/.

[40] Verghese, Ben; Devine, Scott; Gupta, Anoop and Rosenblum, Mendel. Operating System Support for Improving Data Locality on CC-NUMA Computer Servers. Technical report. Computer System Laboratory, Stanford University. 1997. Last Accessed: June 2016. https://courses.engr.illinois.edu/cs533/reading_list/asplos-7-verghese.pdf.

[41] Nyberg, Chris; Koester, Charles and Gray, Jim. Nsort: a Parallel Sorting Program for NUMA and SMP Machines. Version 2.1. Technical Report. Ordinal Technology Corp. and Microsoft Corp. November 1997. Last Accessed: June 2016. http://www.ordinal.com/white/whitepaper.html.

[42] Wikipedia, The Free Encyclopedia. SGI Origin 3000 and Onyx 3000. July 2015. Last Accessed: June 2016. https://en.wikipedia.org/wiki/SGI_Origin_3000_and_Onyx_3000.

[43] Wikipedia, The Free Encyclopedia. NUMAlink. April 2016. Last Accessed: June 2016. https://en.wikipedia.org/wiki/NUMAlink.

[44] Davis, Alan L and Prestor, Uros. The ccNUMA Memory Profiler. Technical report. University of Utah. September 2001. Last Accessed: June 2016. http://www.cs.utah.edu/~ald/pubs/CC-numa.pdf.

[45] Wikipedia, The Free Encyclopedia. IRIX. March 2016. Last Accessed: June 2016. https://en.wikipedia.org/wiki/IRIX.

[46] Taft, Jim. Developments in High Performance Computing: A Preliminary Assessment of the NAS SGI 256/512 CPU SSI Altix (1.5 GHz) Systems. Presentation. NASA Ames Research Center. November 2003. Last Accessed: June 2016. http://people.nas.nasa.gov/~chenze/ECCO/SC03-presentation-part1.ppt.

[47] NASA. NASA Advanced Supercomputing Division. Informational web site. NASA. 2013. Last Accessed: June 2016. http://www.nas.nasa.gov/about/history.html.

[48] DOD. Trusted Computer System Evaluation Criteria. Security policy/Government standard. DOD December 1985. Last Accessed: June 2016. http://csrc.nist.gov/publications/history/dod85.pdf.

[52] Carrier, Brian. File System Forensic Analysis. Book. Addison-Wesley. 2005. ISBN 0-321-26817-2.

[53] SGI. IRIX System Administration I Student Manual & Lab Book. Student Manual. Document No. ISA1-1.4-SM. SGI. May 2004.

[54] SGI. IRIX Admin: Peripheral Devices. Technical manual. Document No. 007-2861-005. SGI. Unknown date. Last Accessed: June 2016. http://csweb.cs.wfu.edu/~torgerse/Kokua/SGI/007-2861-005/sgi_html/index.html.

[55] Wikipedia, The Free Encyclopedia. Silicon Graphics. June 2016. Last Accessed: June 2016. https://en.wikipedia.org/wiki/Silicon_Graphics.

[56] Wikipedia, The Free Encyclopedia. OpenGL Performer. June 2016. Last Accessed: June 2016. https://en.wikipedia.org/wiki/OpenGL_Performer.

[57] Wikipedia, The Free Encyclopedia. OpenGL Multipipe. August 2015. Last Accessed: June 2016. https://en.wikipedia.org/wiki/OpenGL_Multipipe.

[58] SGI. SGI VizServer with NICE Software. Datasheet. SGI. 2015. Last Accessed: June 2016. https://www.sgi.com/pdfs/4430.pdf.

[59] SGI. IRIX Advanced Site and Server Administration Guide. Hardware & systems manual. SGI. 1997. Last Accessed: July 2016. http://rsusu1.rnd.runnet.ru/sgi/advanced/ch8.html.

List of symbols/abbreviations/acronyms/initialisms

55 AA	MBR disk signature
API	Application Programming Interface
CAF	Canadian Armed Forces
CD	Compact Disc
CLE	Canadian Law Enforcement
CPU	Central Processing Unit
CXFS	Clustered XFS
DB	Disk Block
DH	Disk Header
Disc	Refers to an optical disk (i.e., CD, DVD, etc.)
Disk	Refers to a HDD
DND	Department of National Defence
DOD	Department of Defense
DOS	Disk Operating System
DRDC	Defence Research and Development Canada
DVD	Digital Versatile Disc / Digital Video Disc
DVH	Disk Volume Header
dvhtool	Disk Volume Header Tool (system command)
EFS	Extent File System
ESDI	Enhanced Small Disk Interface
Exabyte	$1x10^{18}$ or 1,000,000,000,000,000,000 bytes
GB	Gigabyte ($1x10^{9}$ or 1,000,000,000 bytes)
GPT	GUID Partition Table
GUI	Graphical User Interface
HDD	Hard Disk Drive
hinv	Hardware Inventory (system command)
LE	Law Enforcement
LV	Logical Volume
Lvol	Logical Volume (not the same as LV)
MBR	Master Boot Record

MiB	Mebibyte (2^{20} 1,048,576 bytes)
MIPS	Microprocessor without Interlocked Pipeline Stages
mkfs	Make Filesystem (system command)
mkswap	Make Swap (system command)
NAS	Network Attached Storage
NASA	National Aeronautics and Space Administration
NUMA	Non-Uniform Memory Access
OpenGL	Open Graphics Library
PC	Personal Computer
PROM	Programmable Read-Only Memory
prtvtoc	Print VTOC (system command)
R&D	Research & Development
RAID	Redundant Array of Inexpensive Disks
RCMP	Royal Canadian Mounted Police
RPM	Revolutions Per Minute
SAN	Storage Area Network
SASH	Stand Alone Shell
SCSI	Small Computer System Interface
SGI	Silicon Graphics Inc.
SMP	Symmetric Multiprocessing
SSI	Single System Image
TiB	Tebibyte (2^{40} or 1,099,511,627,776 bytes)
VH	Volume Header
VOLHDR	Volume Header
VR	Virtual Reality
VTOC	Volume Table of Contents
x64	64-bit Intel-based x86 Processor
XFS	Extents File System
xfslog	XFS Log Journal
XLV	XFS Logical Volume
XVM	XFS Volume Management

DOCUMENT CONTROL DATA

(Security markings for the title, abstract and indexing annotation must be entered when the document is Classified or Designated)

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.)

DRDC – Valcartier Research Centre
Defence Research and Development Canada
2459 route de la Bravoure
Quebec (Quebec) G3J 1X5
Canada

2a. SECURITY MARKING (Overall security marking of the document including special supplemental markings if applicable.)

UNCLASSIFIED

2b. CONTROLLED GOODS

(NON-CONTROLLED GOODS)
DMC A
REVIEW: GCEC APRIL 2011

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.)

Forensic analysis of SGI IRIX disk volume

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.)	6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.)	6b. NO. OF REFS (Total cited in document.)
July 2016	40	39

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.)

Scientific Report

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

DRDC – Valcartier Research Centre
Defence Research and Development Canada
2459 route de la Bravoure
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.)
31XF20 MOU RCMP "Live Forensics"	

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.)	10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)
DRDC-RDDC-2016-R127	

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.)

This report examines the partition structures of SGI IRIX disk and optical media, from the perspective of digital forensics. To date, these structures are poorly documented. Should investigators encounter such systems and media, they may have difficulty obtaining meaningful information and evidence from such devices. This report hopes to provide sufficient information to aid investigators with respect to SGI's proprietary UNIX environment.

Le présent rapport examine les structures de partition des disques et des supports optiques IRIX de SGI dans une perspective d'expertise judiciaire en informatique. À ce jour, ces structures sont peu documentées. Si un enquêteur devait se heurter à des systèmes et des supports de ce type, il pourrait avoir des difficultés à obtenir de l'information et des preuves valables. Le rapport, souhaitons-le, fournit suffisamment d'information pour aider les enquêteurs relativement à l'environnement UNIX propre à SGI.

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.)

Data carving; Data recovery; Digital forensics; Disk; Disk partition; Disk Sector; Disk slice; Extent File System; EFS; Extents File System; Fdisk; File system; Filesystem; Forensics; Hard disk drive; Hard drive; HDD; IRIX; Master Boot Record; MBR; Operating system; OS; Partition; SASH; SCSI; SGI; Slice; UNIX; Volume; Volume Header; VTOC; XFS