Recovering virtual machines from lost VMFS partitions

by Washington Almeida and Wellington Rodrigues

Recently, whilst having a meeting with my partner “UTI dos Dados” and a big customer, when we were discussing the options for recovering some VMware virtual machines after its server had been formatted inadvertently, I decided to contact the eForensic team to share our experience using the tool called vmfs-tools that was used to completely recover the environment. As it is not a common situation the technicians experience, we believe this article can bring some help for professionals that may come to face the same scenario.

Although our work was focused on the recovery of customer data, the entire preservation process discussed in the article "Deep Dive into digital forensic case management" in the March 2017 edition ISSN 2300 6986 was applied, as this is one of the mandatory processes in our forensic activities. In this article, we share with the eForensic reader our experience with the vmfs-tools for recovering VMFS partitions and the data contained therein.

About “UTI dos Dados” company:

The company “UTI dos Dados” (http://www.utidosdados.com.br/) specializes in data recovery, which serves customers throughout the Brazilian territory. They are the pioneers in Data Recovery of Monolithic chips.

The company's ability goes beyond data recovery in internal notebook HD, all in one PCs, external HD, server, RAID, NAS, pendrive, memory card, virtual machines and smartphones.

With a highly specialized staff in electronics and first-line equipment, including Class 100 Clean Room, the recovery data and repair work usually surpasses the expectations of its clients.

The importance of the Plan

Customers generally come to us expecting an immediate action from our side, and we understand their timing, as their business may depend on our work.

However, a lesson learned from the classroom of the Law and Information Technology specialization course at the Polytechnic School of USP, the planning and preservation process should be the first part of our work.

Then, taking into account this important requirement, we need to technically understand the scenario in order to plan the most appropriate action for the client, which involves not only the technical issues but also the costs associated with the problem.

The customer disk status

When having a first look into customer RAID disks, we could see four volumes on its partition. This was the situation found when customer had formatted the disks inadvertently.

Okay, there's not much to do here. So it's time to plan our activities to proceed to the environment recovery as we know vmfs partitions are much larger than the volumes shown in Figure 1.

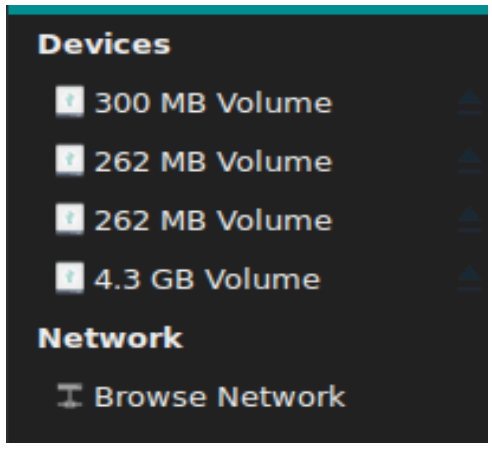


Figure 1 - Customer RAID disks status

The Plan

One of the few tools that support vmfs partitions is the UFS Explorer developed by SysDev Laboratories, which is one of the world leaders in Data Organization and Data Recovery researches. The cost of the tool is about €799.95 and although we have acquired this tool earlier for other purposes, we understand the eForensic reader can manage for recovering the VMWare data with another tool called vmfs-tools, which would avoid this cost.

So, the plan that ensures the preservation of data is summarized in the following lines.

1. Clone the server RAID;
2. Generate the cloned server RAID hash value;
3. Image cloned server RAID disks to generate a RAW image file;
4. Generate the RAW image file hash value;
5. Mount the RAW image file;
6. Identify the deleted partitions in the mounted RAW image;
7. Use vmfs-tools to work into VMFS deleted partition;
8. Access the VMWare machine folders;
9. Restore the VMWare machines.

Once we have defined the plan to win our challenge, let us get a deep dive into each phase to explore them in more detail.

Phase 1 - Clone the server RAID disks

Firstly we have to clone the disks. There are several resources of hardware for this purpose and the equipment used by "UTI dos Dados" company for this task was the FR-100 disk duplicator.

Figure 2 shows the cloning process in action. Note that the source disk is connected in the READ-ONLY interface. This detail is fundamentally important to the preservation process.

Remember that the SATA controller has a BIOS that records all the hard disk operating information. So if this feature is not observed in the process, an experienced engineer may question the evidence during a court proceeding and inevitably invalidate the evidence.Thus, independently, whether you are working on a lawsuit or not, document all phases of the work with photos, operating procedures and all steps used in forensic activity.

Figure 2 – Cloning the disks

Phase 2 - Generate the cloned server RAID disks hash value

Once the disk cloning process is generated, it is equally important to generate the hash results for both disks to ensure that they are exactly the same.

The purpose of generating MD5 and SHA-1 hashes is to ensure that the proof is not altered or modified in each of the subsequent phases from which it was generated. The checksum of the hashes is a digital signature resulting from the use of an algorithm (MD5/SHA-1) on the test media and on the target media, which will result in a unique identification and that can be checked at any time. If there is any change to the proof media, this signature will change and will not match the original signature. Therefore, if we do a sum of checks of a test media before and after its duplication and, if the results are maintained, we can affirm that there were no changes of state in the media of questioned evidence,then if for some reason these results are different, there is a serious problem in the duplication process and it becomes mandatory to redo the cloning activity.

In Brazil (and in the world), this concern aims to comply with Art. 170 of the Code of Criminal Procedure, which states that "In laboratory tests, the experts will keep enough material for the

possibility of a new forensic procedure." Thus after identifying the physical disks, we generate their hashes with the following command line:

```
┌─[wash@parrot]─[~/]
└──╼ $md5sum /dev/sdb
2e6d9d8eaf17abe4ec1982b7475e7856
For the source disk. And for the cloned disk:
┌─[wash@parrot]─[~/]
└──╼ $md5sum /dev/sdc
2e6d9d8eaf17abe4ec1982b7475e7856
```

The hashes results for both source and destination disks is exactly the same as expected. They are also the same for the command line **sha1sum /dev/sdb** and **sha1sum /dev/sdc**.

Phase 3 & 4 - Image cloned server RAID disks to generate a RAW image file

Forensic expert work must be performed on the RAW image generated from the cloned disk. The first disk clone is the source of the evidence.

So before launching the command line to generate the RAW image, it is important to check the size of disk sectors. The command line **fdisk -l** will bring us the disk information including the block size as shown below (in red color):

Units: sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

This is an important point because very often we are called by engineers and companies to analyze and answer why a disk image seems to be corrupted.

As storage densities have increased dramatically over the years, one of the most elemental aspects of hard drive design, the logical block format size know as a sector, has remained constant. However, hard

drive companies are migrating away from the legacy sector size of 512 bytes to a larger and more efficient sector size of 4096 bytes, generally referred to as 4K sectors.

Putting in a nutshell the new standard makes the move to a 4K-byte sector, which essentially combines eight legacy 512-byte sectors into a single 4K-byte sector.

So let's assume that the size of the sector was 4k instead of 512 bytes and we were going to create the image with the defaults, in this case 512 bytes. If it was the case the engineer should be using the bs=4096 instead of bs=512, otherwise the engineer may experience problems when manipulating the image for recovering data.

Thus, the following command line was used to clone our image RAW file:

```
dcfldd bs=512 if=/dev/sdb of=/media/wash/SAMSUNG/250gb.dd hash=md5,sha1
hashlog=/media/wash/SAMSUNG/ImgHashValues
```

Where:

- **dcfldd** - This is an enhanced version of dd for forensics and security which is a copy, converting and formatting a file according to the options.
- **bs=512** - force ibs=512 and obs=512 (ibs reads 512 bytes at a time and obs writes 512 bytes at a time);
- **if=/dev/sdb** - read from FILE instead of stdin (in this case the FILE means /dev/sdb);
- **of=/media/wash/SAMSUNG/250gb.dd** - write to the FILE instead of stdout, in our case the output file is named 250gb.dd;
- **hash=md5,sha1** - The default algorithm for hash option is md5. To select multiple algorithms to run simultaneously we enter the names in a comma separated list;
- **hashlog=/media/wash/SAMSUNG/ImgHashValues** - send MD5/SHA1 hash output to FILE named ImgHashValues instead of stderr.

So the command line above has generated the RAW image file named 250gb.dd and Phases 3 and 4 have concluded since the hashes MD5 and SHA1 planned for phase 4 were generated in the same command line by using the clause hash=md5,sha1.

Phase 5 - Mount the RAW image file

Now that we have already created the RAW image file, observing the preservation process, we are ready to mount the virtual volume of it.

We opt to use losetup to mount the RAW image into /dev/loop using the following command line:

```
sudo losetup -Pf --show -v /media/wash/SAMSUNG/250gb.dd
```

Where:

- **losetup** is used to set up and control loop devices. It associates loop devices with regular files or block devices, to detach loop devices and to query the status of a loop device. If only the loop_device argument is given, the status of the corresponding loop device is shown. The losetup command returns 0 on success, nonzero on failure. When losetup displays the status of a loop device, it returns 1 if the device is not configured and 2 if an error occurred that prevented losetup from determining the status of the device.
- **-Pf** - Force the kernel to scan the partition table on a newly created loop device. The clause **f** just after **-P** finds for the first unused loop device. If a file argument is present, use this device. Otherwise, print its name.
- **--show** - Display the name of the assigned loop device if the -f option and a file argument are present.
- **-v** – Turn verbose mode on.

Thus the return for our losetup command line was **/dev/loop0** which means success on defining the loop device that has been defined by the system with the same named **/dev/loop0** which is the first unused loop device.

Now that we have **/dev/loop0** ready for use, Phase 5 has been concluded. Let us go ahead and get started into Phase 6.

Phase 6 - Identify the deleted partitions in the mounted RAW image:

Now our interest is to identify the partition where VMWare virtual machines were residing. Then we use the fdisk command line to get more information about the partitions as follows:

```
sudo fdisk -lu /dev/loop0
```

The fdisk command is used to manipulate the disk partition table. The clauses **-lu** tells fdisk to:

- **-l** - List the partition tables for the specified devices and then exit. If no devices are given, those mentioned in /proc/partitions (if that file exists) are used.
- **-u** - When listing partition tables, the **-u** clause shows sizes in 'sectors' or in 'cylinders'. The default is to show sizes in sectors.

So the output for the **fdisk -lu /dev/loop0** command line is shown below:

```
Disk /dev/loop0: 232.9 GiB, 250059350016 bytes, 488397168 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: E5C7B4C5-3E39-4B83-BC1B-08239765BEAF
Device     Start        End      Sectors      Size     Type
/dev/loop0p1   64       8191     8128     4M       EFI System
/dev/loop0p2   1843200  10229759 8386560  4G       Microsoft basic data
/dev/loop0p3   10229760 389545950    379316191    180.9G   unknown
/dev/loop0p5   8224     520191   511968       250M     Microsoft basic data
/dev/loop0p6   520224   1032191  511968       250M     Microsoft basic data
/dev/loop0p7   1032224  1257471  225248   110M     unknown
/dev/loop0p8   1257504  1843199  585696   286M     Microsoft basic data
```

Although we have two partitions not recognized by fdisk utility we can easily perceive the partition defined as **/dev/loop0p3** is where the VMware Virtual Machines must be residing in and the partition **/dev/loop0p7** is the Vmware core partition. We know that because the customer let us know about it in

our meeting and also because the partition **/dev/loop0p3** is is the only partition with enough space to support virtual machines as it usually hosts big files. Let us investigate the **/dev/loop0p3** using the fdisk again.

```
sudo fdisk -lu /dev/loop0p3

Disk /dev/loop0p3: 180.9 GiB, 194209889792 bytes, 379316191 sectors

Units: sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Phase 7 - Use vmfs-tools to work into VMFS deleted partition:

The only way to access data into a VMFS partition is to make use of a tool that recognizes this type of partition. At this point we want to share with the eForensic reader an option of a package available in Linux Debian repositories named vmfs-tools; it is extremely efficient and can be operated at zero cost.

VMFS stands for Virtual Machine File System, and the vmfs-tools are in an Open Source Virtual Machine File System (VMFS) Driver.

This driver enables read-only access to files and folders on partitions formatted with the Virtual Machine File System (VMFS). VMFS is a clustered file system that is used by the VMware ESX hosts to store virtual machines and virtual disk files.

After it's installed, the VMFS driver comes with command line interface (CLI) tools to mount and analyze VMFS volumes. The VMFS driver was tested on Linux and Windows based hosts, but should work on any platform that supports Java.

Thus the VMFS volumes can also be browsed or mounted using a WebDAV client.

In order to install vmfs-tools in a Linux Debian based machine, we use the following shell command line:

```
sudo apt-get update && apt-get install vmfs-tools
```

Alternatively, we can clone vmfs-tools from the GitHub repository located at URI https://github.com/glandium/vmfs-tools using the following shell command line:

```
git clone https://github.com/glandium/vmfs-tools.git
```

This command line above will install the package inside the current folder where the command is launched.

Once we have the vmfs-tools installed in our environment, we have to use it in order to use the volume of our interest. To do that we use the following shell command line:

```
sudo vmfs-fuse /dev/loop0p3 /home/wash/mnt/vmfs
```

Where:

- **sudo** – SuperUser do, or, do as a SuperUser. The root user is the SuperUser;
- **vmfs-fuse** - vmfs-tools evolved to add more VMFS features and supports read only VMFS mounts through the standard Linux VFS and the FUSE framework. FUSE stands for Filesystem in UserSpacE and it is a software interface for Unix-like computer operating systems that lets non-privileged users create their own file systems without editing kernel code. This is achieved by running file system code in user space while the FUSE module provides only a "bridge" to the actual kernel interfaces;
- **/dev/loop0p3** - The partition where VMWare machines resides;
- **/home/wash/mnt/vmfs** - the folder from where we want to access VMWare machine's data.

Phase 8 - Access the VMWare machine folders

Now that we have the vmfs-tools installed, let us check the partition's status again using the following command line:

```
sudo fdisk -lu /dev/loop0

Disk /dev/loop0: 232.9 GiB, 250059350016 bytes, 488397168 sectors

Units: sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disklabel type: gpt
```

```
Disk identifier: E5C7B4C5-3E39-4B83-BC1B-08239765BEAF

Device        Start        End      Sectors      Size     Type
/dev/loop0p1  64       8191     8128     4M       EFI System
/dev/loop0p2  1843200  10229759 8386560  4G       Microsoft basic data
/dev/loop0p3  10229760 389545950    379316191    180.9G   VMware VMFS
/dev/loop0p5  8224     520191   511968       250M     Microsoft basic data
/dev/loop0p6  520224   1032191  511968       250M     Microsoft basic data
/dev/loop0p7  1032224  1257471  225248   110M     Vmware VMKCORE
/dev/loop0p8  1257504  1843199  585696   286M     Microsoft basic data
```

At this time we can see both actual data as shown in figure 1 and also the lost VMware partitions. As the partitions are recognized by vmfs-tools, we can explore the content of it. So let us list the content of **/dev/loop0p3** with the shell command line below:

```
sudo ls /home/wash/mnt/vmfs
debian-8.7.1-i386-CD-1.iso
Server
W_XP_01
W_XP_03
W_XP_05
W_XP_10
W_XP_02
W_XP_04
W_XP_09
```

Bingo, we are able to access folders into deleted VMFS partition. Let us have a look inside the Server folder.

```
sudo ls -l /home/wash/mnt/vmfs/Server
-rw-r--r-- 1 root root     27 Apr  4 18:14 Server-788dab49.hlog
-rw------- 1 root root69793218560 Apr 22 11:55 Server-flat.vmdk
-rw------- 1 root root8684 Apr 22 11:55 Server.nvram
```

```
-rw------- 1 root root491 Apr 11 17:41 Server.vmdk
-rw-r--r-- 1 root root0 Oct 22  2013 Server.vmsd
-rwxr-xr-x 1 root root4088 Apr 22 11:55 Server.vmx
-rw-r--r-- 1 root root364 Apr  4 18:18 Server.vmxf
-rw-r--r-- 1 root root514483 Apr  4 17:54 vmware-66.log
-rw-r--r-- 1 root root249824 Apr  4 17:56 vmware-67.log
-rw-r--r-- 1 root root250014 Apr  4 17:59 vmware-68.log
-rw-r--r-- 1 root root1195724 Apr  4 18:14 vmware-69.log
-rw-r--r-- 1 root root191602 Apr  4 18:15 vmware-70.log
-rw-r--r-- 1 root root591020 Apr 11 17:41 vmware-71.log
-rw-r--r-- 1 root root1917847 Apr 22 11:55 vmware.log
```

Phase 9 - Restore the VMWare machines

Finally, as root, I can make a simple copy of the contents found in Phase 8 as following:

```
cp -r /home/wash/mnt/vmfs/ /media/wash/SAMSUNG/vmfs/
```

Where:

- **cp** - Linux command line to copy files and directories;
- **-r** - copy directories recursively;
- **/home/wash/mnt/vmfs/** - mounted /dev/loop0p3 partition;
- **/media/wash/SAMSUNG/vmfs/** - external volume where I want to save the data.

Summary

As we could see, any digital evidence is fragile enough to be easily destroyed by improper handling.

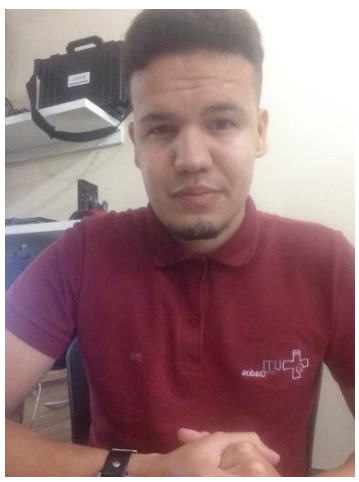
However, by using appropriate techniques, it is possible to provide data recovery under the most adverse circumstances. UTI dos Dados is one of the very few companies that surprises for its technical competence in the recovery of the most varied types of media. Many media that arrive at “UTI dos

Dados" had the diagnosis that nothing else could be done, and the company competently recovered the data in the face of such diagnoses.

The challenges in digital forensics are immense by virtue of the constant and rapid transformation in the assets of the information technology environment, such as hardware and software, as well as open and proprietary technology solutions. Any kind of data that is considered irrecoverable is worth the effort and the dedication of competent professionals to show that even in critical situations it is possible to bring a positive solution to your client, and on many occasions we have the excellent tools available at no cost that can help in this job. The commitment to perception that may always be possible for recovering data is fundamentally important for the justice.

About the Authors

Washington Almeida is an Electronic Engineer specialized in Digital Forensics and Cyber Security with more than 25 years of experience in the Information Technology and Engineering areas, working for large companies in the sectors as Engineering, Information Technology, Consulting, Chemical and Mining. Experienced professional with Cisco and Microsoft MCSE certifications acts as Digital Forensics with in-depth knowledge of computers hardware, network technologies, telephony, programming, data communication protocols and a vast of information security knowledge with a set of skills known by ethical hackers where this knowledge base is fundamental to assist the Justice.
Website: www.washingtonalmeida.com.br



Wellington Rodrigues is a professional who has advanced knowledge in data recovery, CEO of UTI Dos Dados , recovers data from all types of devices including Hds, Sdds, Raids, Virtual Machines, smartphones, pioneers in data extraction of Flash memories and Monolithic chips in Brazil, Including works with microsoldering. Professional has certifications HP is director of a company specialized in notebooks serves all Brazilian territory for advanced repairs of HP and Compaq.
Website: www.utidosdados.com.br