# Feasibility of Digital Forensic Examination and Analysis of a Cloud Based Storage Snapshot

Sameera Almulla, Youssef Iraqi
College of Electrical and Computer Engineering, Khalifa University of Science Technology and Research, UAE
sameera.almulla@kustar.ac.ae
youssef.iraqi@kustar.ac.ae

Andrew Jones
Department of Computing, Engineering and Science
University of South Wales, UK
Edith Cowan University, Australia
andy.jones@southwales.ac.uk

*Journal of Digital Information Management*

***ABSTRACT:*** *With the increasing use of digital technologies and the subsequent escalation in crimes that involve these technologies, the role of digital forensics has become increasingly more important. This is mainly because both businesses and individuals have become more dependent on digital devices and Internet services. In addition, the ease and availability of large-scale storage and massive computing resources with relatively low cost have made cloud computing an attractive option for many users and organizations.*

*In the cloud environment, most of the time investigators will end up with only partially acquired evidence that has been provided by the Cloud Service Provider (CSP). Therefore, researchers in the field of cloud forensics need to move away from insisting on acquiring all data -as has historically been the case in computer forensics- and yet still be able to prove the accuracy, sufficiency, and soundness of partially acquired data. Virtualization is considered to be one of the main pillars in providing cloud services. In some cases, investigators might end up having to rely on suspect Virtual Machine (VM) snapshots in the form of memory dumps and user activity logs. Then, in these cases, the primary challenge is to analyze these memory dumps without altering the evidence. In this paper, after assessing static and live forensics tools in examining cloud-based snapshot, we propose a forensic procedure based on the National Institute of Standards and Technology (NIST) model to examine the private cloud VM snapshots (e.g. XenServer). Moreover, we tested snapshots using existing digital forensic tools and were able to successfully acquire data without the need to transform the snapshot files.*

**Subject Categories and Descriptors**
**[B.3 Memory Structures]: Virtual memory**; **[E.2 Data Storage Representations]:** [D.4.6 Security and Protection]: **[H.2 Database Management]:** Security, integrity, and protection

**General Terms**
Cloud Storage, Computer Forensics

**Keywords:** Cloud Forensics, Virtual Machines, Private Cloud

**Received:** 12 November 2016, Revised 4 December 2016, Accepted 10 December 2016

## 1. Introduction

With the scalability and cost-effectiveness of cloud

services, the associated risks are considered to be a risk that individuals and organizations are willing to take. The growing trend, not only in providing cloud-based services but also the adoption of these services has led to an increasing need for both security and digital forensic research.

As stated in [1], the term "Cloud Forensic" was introduced in 2011, and since then, several research directions have been proposed for further studies in this field. In crimes where the evidence is located in the cloud, acquiring forensically sound evidence is not always feasible for several reasons; (1) the geographically dispersed storage of evidence in a cloud environment, (2) the impact of each jurisdiction laws and procedures, and (3) the lack of standardization in the Service Level Agreement (SLA) between the client and the Cloud Service Provider (CSP). Even if the CSP complies with the law enforcement agencies in their respective jurisdictions, this may be a costly and time-consuming exercise, given the massive amount of data that belongs to the shared pool of storage.

Virtualization is considered as a main enabling technology for cloud-based services [1]–[3]. According to a Cisco survey [4], about 65 % of small businesses (50-100 employees) are currently using virtualization technology, and that jumps to 79 % for medium-sized companies (100-500 employees). Among the companies that were already using virtualization, 96 % believed that they were gaining an advantage and this has led to a growing interest in adopting virtualization technology to provide services. These statistics along with the fast pace with which the technology is moving, increase the probability of encountering a virtual environment during a digital investigation.

Using VM to create contained environments either to examine suspect devices or for malware isolation has been practiced for several years. However, the virtual environments themselves have become a target for investigation. Therefore, the existence of snapshot technology can be considered as an opportunity in the course of an investigation.

Snapshot technologies present several advantages; such as (1) the ability to be seized offline, (2) minimal disturbance to business continuity during an investigation, and (3) the encapsulation of both memory and storage. However, the main drawbacks are (1) the snapshot is a Point in Time (PIT) copy of data, hence it may not contain the most up-to-date data, (2) whether the current snapshot methodologies satisfy the digital forensic requirements, and (3) whether an investigator can acquire the VM snapshots.

The main cloud service providers such as Rackspace and Amazon are using XenServer to provide Infrastructure-as-a- Service (IaaS). Therefore, in this paper, Citrix XenServer will be studied as a test platform. Even though the XenProject is an open source software under GNU Public License v2, the Citrix XenServer v6d, various system drivers and user tools are still owned by Citrix and remain proprietary[1]. Moreover, CSPs that adopt the XenServer or XenProject may customize the source code based on the organizational needs. In the case of an incident, the investigators will end up investigating a VM snapshot that is equivalent to a proprietary virtualization product.

Due to lack of technical studies that can help in the examination of Xen-based digital data, in this paper, the contributions are;

• To present a practical assessment whether
– the requirements of the used snapshot method are sufficient and necessary to generate forensically sound evidence, and
– the existing digital forensics tools are sufficient to examine snapshots.

• To analyze snapshots without the need to change the file format. Changing the file format might result in data loss, and this will affect the evidence integrity.

• To propose a digital forensic procedure of cloud based snapshots.

• To acquire evidence from VM snapshots without the need to recreate the virtual environment.

The paper is organized as follows; Section 2 discusses the related digital forensic research in the virtual environments followed by a discussion on the main concepts in Section 3. Sections 4 and 5 present the structure of Citrix XenServer snapshot and the used terminologies respectively. Section 6 discusses a set of cloud-based snapshot requirements to produce forensically sound evidence. The assessment of existing digital forensic tools in examining snapshots is discussed in Section 7. The proposed digital forensic procedure is presented in Section 8. A hypothetical case scenario with a practical assessment of the proposed procedure is discussed in Section 10 followed by a discussion on the results in Section 11. Finally, this paper concludes in Section 12.

## 2. Related Work

In a cloud environment, a snapshot can be defined as a PIT copy of the storage and its meta-data as discussed in [3], [5]– [10].

Huseinovic *et.al.* [6] analyzed two virtual environments, namely Oracle VirtualBox and VMware workstation. The data content of snapshot files is first transformed using a thirdparty tool and then analyzed. The snapshot file transformation might result in an evidence contamination. Examiners need to maintain a chain of custody throughout the investigation process.

[1]XenServer, (2016), http://xenserver.org/overview-xenserver-open-sourcevirtualization/source-code.html

Zhang *et.al*. [7] and Guangqi *et.al.* [8] proposed a kernelbased virtual machine (KVM) approach to acquire both data and VM meta-data. In their work, the key aspect in acquiring VM-related information was to have access and control as a VM host. Similarly, Saibharath and Geethakumari [9] analyzed OpenStack snapshot files using their proposed software. To retrieve the snapshot content and its meta-data the examiners would need to have host privileges, which might not be possible in the general case. The current practice is that the investigators communicate with the CSPs to provide the required logs and memory dumps including VM snapshots. On the other hand, the authors in [11] discussed the process of memory acquisition without any CSP interaction. The research focused on the self-service characteristics of the cloud environment. In [11], the author studied KVM storage and memory images in OpenStack.

Another limitation of existing techniques is the need to construct the virtual environment (either KVM or VMware) to recreate the VM using its data and meta-data. Diane *et.al.* stated in [12] that there might be cases where the only way to examine the virtual environment is by using the tool that created the environment itself.

## 3. Background

### 3.1 Storage Snapshot Methods

A storage stack consists of many hardware and software components which provide the storage for the applications that are running on the host operating system. Besides the different methodologies to implement the snapshot, it can be implemented either at the hardware layer or at the software layer. A storage snapshot can be defined as a process of recording the state of storage at a PIT and preserving that snapshot to restore the data in the event of failure, and for data protection and troubleshooting.

There are two categories of storage snapshots namely controller and software based snapshots [13]. The former is managed by storage vendors such as IBM and EMC and done at the Logical Unit Number level- independently of the file system and the operating system. The latter is performed by the file system, volume manager or a third party that is independent of the underlying storage and hardware.

In general there are six snapshot methods namely, Copy On Write (COW), Redirect On Write (ROW), Split Mirror Redundant Array of Independent Disks (SMRAID), Log Structure File Architecture (LSFM), COW with background copy (COW+), and incremental and Continuous Data Protection (CDP) [13]. In widely adopted IaaS environments such as Xen and VMware, the COW snapshot is the default method that has been used to generate VM snapshots. Hence, the rest of this paper and technical study will focus on the COW method.

### 3.2 Citrix XenServer Architecture Overview

There are two main types of virtualization, software, and hardware. The former is used to emulate a complete computing system to allow a guest operating system to be run e.g. VMplay or Oracle VM VirtualBox. The latter is deployed directly onto the hardware with the aid of hypervisor and used in utilizing the storage and network resources e.g. XenServer. The Hypervisor is a software program which enables the virtualization of the hardware, such as memory, storage, and processors [12].

Hardware virtualization can be extended to a further three categories, namely full-virtualization, paravirtualization and hardware-assisted virtualization; this is categorized based on the techniques used for handling instructions and communications between the guest VM, its applications, and the hardware.

• **Full-virtualization [12]:** Is used to provide a virtual environment that completely simulates the underlying hardware. The hypervisor provides each VM with all of the services of the physical system including a virtual BIOS, virtual network devices, and virtualized memory management. User applications in the guest VM can directly execute instructions through binary translation of Operating System (OS) instructions.

• **Paravirtualization [12]:** Requires modification in the OS kernel. It acts as a bridge -hypervisors system calls to communicate with the hardware- between the application and the processing performed at the hardware level.

• **Hardware-assisted virtualization [12]:** Is a technology that allows the CPU instruction set communication in which the hypervisor runs in a new root level mode. Privilege system calls are initialized to trap the hypervisor. Hence, the use of binary translation as in full-virtualization or the use of the hypervisor calls as in the paravirtualization are no longer needed.

Xen is a large project, comprised of many hundreds of individual software components. In this paper, we examine the Citrix XenServer (for short XenServer) 6.2.0 snapshots. XenServer is a bare-metal hypervisor and uses paravirtualization technology for better performance.

### 3.3 Citrix XenServer Implementation and Specification

The specification of the deployed XenServer is described in Figure 1. There are two methods for VM snapshot acquisition, either using the Command Line Interface (CLI) or client management console. In crime cases, where the CSP is the target of a crime, CLI resultant snapshot might be under investigation. However, in the case where the client is a target, the provided snapshot might be a result of management console.

In this paper, we will highlight both methods with a brief discussion on pros and cons of each. However, the snapshots used in Section 10 have been generated using the management console.

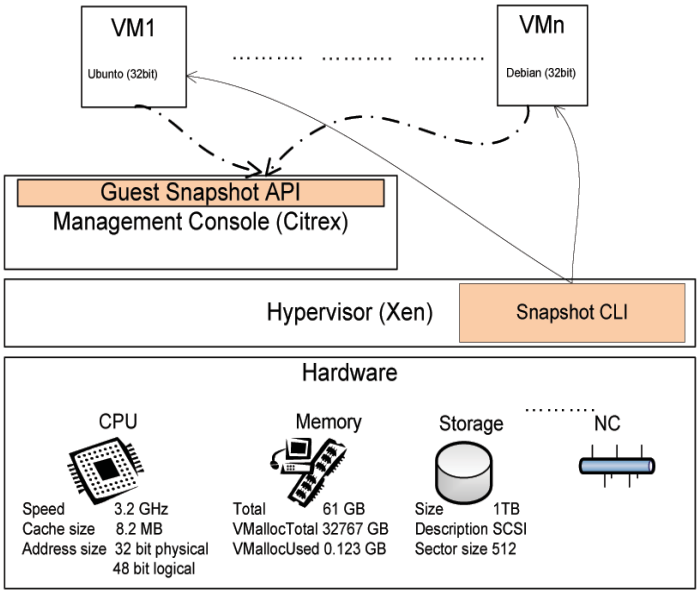


Figure 1. The deployed XenServer specification

### 3.4 Citrix XenServer Snapshot

Snapshot files are unstructured files in comparison to the hard disk image (bit-by-bit copy). Most of the computer forensic tools require a structure description (e.g. file system or hard disk partition scheme) in order to examine the storage and retrieve the data. It is worth mentioning that the VM snapshot can be retrieved using the export and the import features available in the Xen environment. However, the aim is to investigate the feasibility of extracting evidence from snapshot files even if the Xen environment is not available.

## 4. Structure of Citrix Xenserver Snapshot

In Citrix XenServer, the snapshot file consists of a header, a sequence of compressed files and the End-Of-File marker as depicted in Figure 2. The header also known as Open Virtual Application (OVA) is an xml file that consists of the VM meta-data and the meta-data of the subsequent compressed files. Each compressed file consists of the snapshot data with a checksum and a unique identifier.

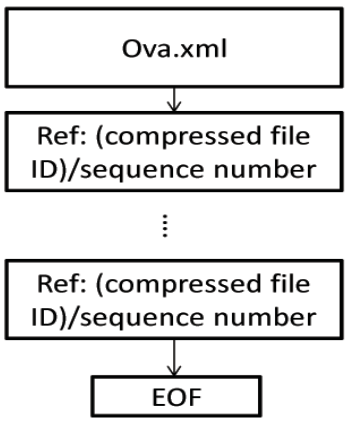


Figure 2. The Structure of Snapshot File

## 5. Xenserver Terminologies

Before proceeding further, the terminologies used in the virtual environment will be explained [14]. A Virtual Hard Disk (VHD), is a virtual storage allocated to each virtual machine once it is created. A collection of VHDs are stored in a Storage Repository (SR). In the case where the user wishes to mount a storage to their VM, that storage first needs to be stored in the SR, and then it can be attached to the required VM.

One of the XenServer features is to image the VHD for backups and to create snapshots. These images are called Virtual Disk Images (VDIs). To uniquely identify the virtual storages and the VM, XenServer allocate a Universal Unique IDentifier (UUID) to each VDI.

On “take snapshot”, the Xen environment creates three copies of VHDs namely; snapshot, active and parent node. Figure 3 depicts these entities and their relationships. When a user performs a write action, the file will be first copied to the child snapshot and then overwrite the file in the active snapshot.

The investigators will end up dealing with the child snapshot as a source of evidence. Next, we present the experimental assessment of the existing digital forensics tools in examining the snapshot contents.

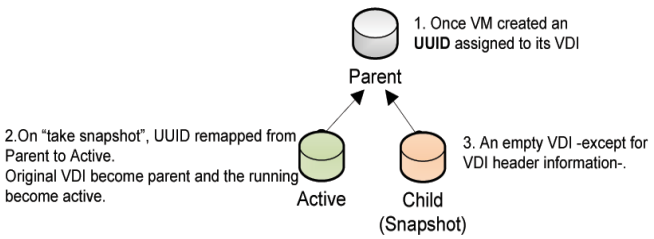


Figure 3. VM and its Snapshot [14]

## 6. Digital Forensic Requirements

Dykstra and Sherman [5] and Stephenson [15] suggested the (1) *integrity* of the snapshot as a main requirement. Furthermore, we mapped the traditional digital forensics evidence requirements to snapshot methods, such as (2) the sufficiency of the *logical acquired* snapshot in producing forensically sound evidence, (3) *independent from original storage*, that is, whether the acquired snapshot storage is sufficient to build the case. For example, an incremental snapshot records the update on the storage from the last snapshot, where the examiners will require original storage and the snapshot for data acquisition, (4) *target storage is wiped* to ensure that the integrity of the snapshot had been preserved during the acquisition, and (5) *zero snapshot latency*, where the snapshot of the desired storage is created with zero or minimal latency. In this paper, we will conduct experimental assessment to check whether the COW snapshot method satisfies these requirements and the results will be discussed in Section 11.

## 7. Assessment of Existing Digital Forensics Tools

The digital forensic procedure can be performed either as static or live. The former deals with the digital evidences that have been acquired off-line, while the latter manages the evidence acquisition in real time. In both cases, the investigators use an appropriate digital forensics tool to acquire forensically sound evidence.

In our experiment, four categories of tools have been used; (1) computer forensic (using Forensic Tookit (FTK) tool [16]), (2) memory forensic (using FTK Imager [16]), (3) file carver (using Scalpel tool [17]), and (4) binary editor (using Hex Editor (HxD) tool [18]).

The main findings are that the content of snapshot files could not be retrieved using FTK and FTK imager. However, both the scalpel and HxD were able to extract the meta-data of the VM along with the meta-data of the content and the data. Table 1 lists the previously discussed tools and their capabilities to retrieve the data from the snapshot.

The Scalpel is designed to retrieve files in the from of a .image file. However, our experiment proved that other binary files content could be extracted using this tool. Snapshot files with the extension .xva can be transformed to .img files. However, further study is needed to ensure the integrity of the snapshot files and to make sure they are not altered due to the transformation process.

As shown in Table 1, some of the tools were able to retrieve traces of the evidence. However, the Scalpel was able to view the image file using an appropriate editor viewer. HxD was able to locate the text file and traces of the content, but the tool does not support viewing evidence using a text or multimedia viewer application.

To overcome the limitations of the existing digital forensic tools in examining snapshots, in the subsequent sections, Digital Forensic Framework (DFF) [19] will be used to assess the proposed digital forensic procedure. DFF is an open source digital forensic tool that is used by professionals and researchers in the forensics field.

## 8. The Proposed Digital Forensic of Cloud Based Storage Snapshot (DF-CBSS) Procedure

In this paper, we propose a Digital Forensic of Cloud Based Storage Snapshot (DF-CBSS) procedure to examine the snapshot files based on the National Institute of Standards and Technology (NIST) model [20]. The proposed methodology consists of identification, integrity verification, on top of the NIST model of collection, examination, analysis and reporting.

• **Identification:** Is a process whereby the main

| Acquisition Tool | VM meta-data | Content meta-data | Content | Content Viewer |
|---|---|---|---|---|
| FTK | NO | NO | NO | NO |
| FTK Imager | NO | NO | NO | NO |
| Scalpel | YES | YES | YES | YES |
| HxD | YES | YES | YES | NO |

Table 1. The Assessment of The Used Tools To Retrieve Snapshot Content

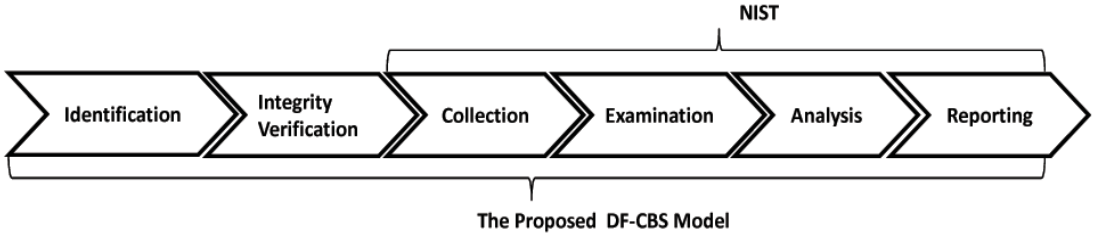


Figure 4. The Proposed DF-CBSS Procedure

components to conduct the investigation are determined, such as (1) the digital forensics requirements that need to be satisfied by the snapshot method, (2) the used virtual environment (e.g. VMware or Xen), (3) the type of the crime, whether CSP was a target or used as a tool to conduct a crime. This will help to identify which approach investigators can follow to acquire VM snapshot either using CLI or management console.

• **Integrity Verification:** Is a process to ensure that the act of acquiring the digital data does not alter the snapshot files.

In XenServer, "xva verify.exe" is provided to verify whether the snapshot is broken or not. "xva verify.exe" can be executed using the Windows cmd and the output is a series of checkpoints performed on the snapshot file (as depicted in Figure 5).

In DF-CBSS, the integrity verification process will take the suspect VM snapshot as an input. The resultant list of checksums of both "xva verify.exe" and DFF tool are then compared to test the integrity of the "xva verify.exe". Figure 6 describes the DF-CBSS workflow of the integrity verification process.

• **Collection:** Is a process of identifying and collecting items that are considered of evidential value [20]. In XenServer, the potential source of evidence is the snapshot file. The snapshot file is comprised of series of compressed segments, and as stated in [21], a collection of evidence in the form of lossless compression is the area of focus in digital investigation.

In computer forensics, examiners aim to collect the existing files (e.g. logs, multimedia, ... etc), deleted files, and hidden files (e.g. that exist in the slack space) along with the time stamps. In this paper, we assess whether the DF-CBSS procedure is capable of collecting the evidence from the VM snapshot. As discussed in Section III-C, the snapshot acquisition might be performed either using CLI or the management console. The former requires further knowledge to extract the correct snapshot file.

Next, we will highlight the basic CLI commands that can be used to collect the desired VM snapshot.

**How to identify a snapshot?** Using CLI command, xe vdi-list sr-uuid=xxx-xxxx-xxxx-xxxx-xxxx params=name-label is-a-snapshot, this will return the active snapshot VDI-UUID [14].

**Where is the snapshot of a VM stored?** Using the CLI command xe vdi-list a list of all VDIs and their metadata

```
0040475 000000000000 ova.xml skipping ova.xml
1048576 000000000000 Ref:29/00000000 has checksum: 71a54eb73b28c6e999bd69cc97e1faead149af38
0000040 000000000000 Ref:29/00000000.checksum Ref:29/00000000 hash OK
1048576 000000000000 Ref:29/00000001 has checksum: 9219c6dfe935097fd2001f1982220fff9059c589
0000040 000000000000 Ref:29/00000001.checksum Ref:29/00000001 hash OK
1048576 000000000000 Ref:29/00000002 has checksum: 632373e8ec70751eb6c2ae9b6215aecb32253469
0000040 000000000000 Ref:29/00000002.checksum Ref:29/00000002 hash OK
1048576 000000000000 Ref:29/00000003 has checksum: 4aa55ea51d42e2a82821a9374a549e67915258f3
0000040 000000000000 Ref:29/00000003.checksum Ref:29/00000003 hash OK
1048576 000000000000 Ref:29/00000004 has checksum: 37e112a4e3d185bf4a62c540c00d9226057e73b3
0000040 000000000000 Ref:29/00000004.checksum Ref:29/00000004 hash OK
```

Figure 5. Excerpt of Output After Running xva_verify.exe On A Suspet VM Snapshot

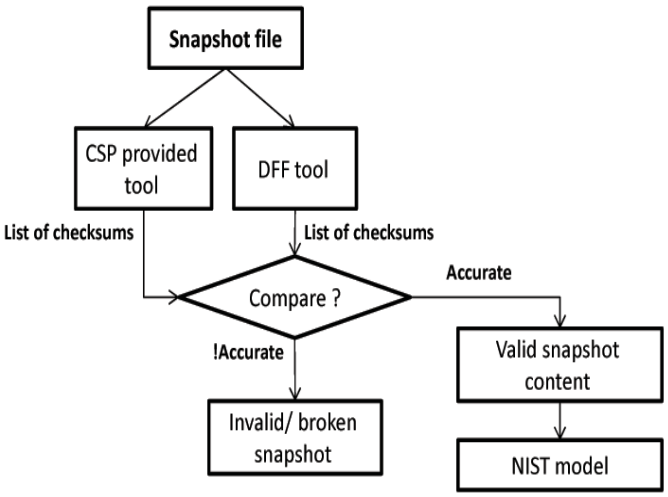


Figure 6. Snapshot File Integrity Checker

| Name | NEW VM Snap1 |
|---|---|
| Description | P:Parent<br>S:Snapshot |
| VM-UUID | P:0ea79e95-8244-7bdf-e0c3-af6fa1e68137<br>S:cfa766ff-d5ea-3deb-cf3f-2402ad865274 |
| VDI-UUID | P:c5680b38-8e89-4d11-bb51-a4ebf167793f<br>S:15171b39-68f6-442e-8552-ella8ea05074 |
| VHD-UUID | P:a48fe976-95e8-4c1b-f837-c7abeea3daf7<br>S:8c9c2aaf-e82a-e599-135f-5d4109b795f4 |
| SR-UUID | P:0913ca53-6629-e96d-7630-d31735b3691a<br>S:0913ca53-6629-e96d-7630-d31735b3691a |
| Virtual-Memory-size | P: 8 GB<br>S: 8 GB |
| Used | P: 36 MB<br>S: 21 KB |
| CLI command | P:xe vbd-list vm-name-label=”NEW VM”<br>P:xe vdi-list uuid=“VDI-UUID”<br>S:xe vbd-list vm-name-label=“snap1”<br>S:xe vdi-list uuid=“VDI-UUID” |

Table 2. Comparison Between VM And Its Snapshot Meta-Data: An Example

can be viewed.

**What are the parameters used to identify the VM?** There are three main parameters used to identify a VM and its storage. VM-UUID, VHD-UUID and VDI-UUID are three distinct IDs assigned to a single VM. Furthermore, an SRUUID is assigned to the shared storage repository where these three entities are stored.

**Used snapshot methodologies.** XenServer uses the COW snapshot methodology.

**What is the content of the snapshot storage?** The snapshot consists of configuration information such as metadata and raw data stored on the VHD. A comparison between the VM and snapshot meta-data is shown in Table 2 for an example scenario.

It is clear that the meta-data of the snapshot storage “snap1” is different than that of the parent VM “NEW VM”. Another observation is that both snapshot and the parent are assigned a same size VHDs; however the used capacities are different -given that parent is newly installed VM- and the snapshot had been taken before even logging into the parent VM.

To uniquely distinguish the parent, active and child storage snapshot, XenServer assigns a different VM-UUID for each. The only common parameter is the SR-UUID, which represents the shared pool of storage in XenServer.

• **Examination:** Is a process of assessing and extracting the relevant pieces of information from the collected data. This can be achieved using forensic tools. Some of them are open source, some proprietary, and others are available only to law enforcement institutions. However, there are many standard tools used by system administrators for day-to-day work, which were not primarily designed as forensic tools, but can be used as such.

• **Analysis and reporting:** Is a process to study and analyze the data to draw conclusions from it [20]. It also relates to whether an investigator can reconstruct the timeline of the suspect VM user’s activity.

Finally, the process of preparing and presenting the information resulting from the analysis phase is considered for reporting.

## 9. Investigation Case Scenario

Diane et.al. in [12] discussed several investigation case scenarios for virtualization and the cloud. One similar case is where an investigation was required to examine Xen VM artefacts. In this case, an organization uses a private cloud to host VMware based on the Windows 7 Operating System. An employee left the company to join a major competitor, and the organization requests that an investigator examines the ex-employee’s external hard drive. During the course of investigation, .xva files were found in unallocated space. The organization asked the investigator to provide a report of the finding including any recovered files, deleted files and VM meta-data.

The primary objectives of deploying DF-CBSS are (1) to retrieve desired evidence without altering the source of

the evidence (in this case without changing the file format), and (2) whether the investigator can retrieve the evidence without the need for the tools that were used to build the virtual environment.

## 10. Using DF-CBSS

### 10.1 Identification

Since the .xva file (size 3.80 GB) was provided, the investigation target the XenServer virtual environment. As discussed earlier, XenServer adopts COW to create a VM snapshot. COW first creates a VDI and then copies all changes made on the original VDI to the child VDI. As stated in Section 6, we identify the digital forensics requirements that need to be satisfied by the COW method.

### 10.2 Integrity Verification

The organization provides a XenServer snapshot file named "snap.xva". The file integrity is verified using the approach discussed in Figure 6. The result was as depicted in Figure 7, where the "snap.xva" is a valid snapshot file. While the snapshot file size is scalable, the execution time of the integrity verification will increase with the increase in the corresponding snapshot file size.

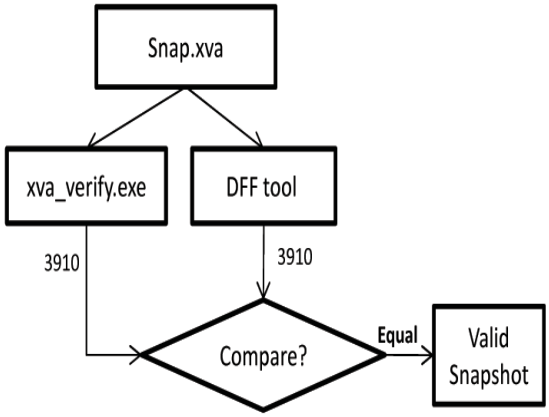


Figure 7. Snapshot File Integrity Checker Result

### 10.3 Collection

As discussed in [3], a snapshot can be taken using the CSP API (e.g. Citrix XenCenter API ) or at the hypervisor level as depicted in Figure 1. From a digital forensics perspective, there is a tradeoff between taking the snapshot at each layer. For example, a snapshot via the CLI gives more accessibility to the virtual storage and the VM's meta-data in comparison to the API [7]. However, an API based snapshot may guarantee that no privacy violation of legitimate users data occurs while it is more difficult to maintain using CLI commands.

### 10.4 Examination

***1) Meta-data:*** In the examination process, all relevant data needs to be retrieved. The organization requested the examiners to identify the snapshot file content and, using the DFF the examiners retrieved the following data from the suspect snapshot. A sample of the extracted meta-data of the suspect hypervisor, VM and snapshot are listed in Table 3, 4, 5 respectively.

| Hyper meta-data | Description |
|---|---|
| Vendor | Citrix XenServer6.2.0 - XAPI |
| H-name | lab 112-6-xen |
| H-IP(address) | null |
| CPU Info | CPU-count = 4 socket-count =1 vendor= GenuinIntel speed =3200.110 model-name=Intel(R) Xeon(R) |
| BIOS vendor and version | Hewlett-Packard and 786 G3 v03 07 |
| System manufacturer | Hewlett-Packard |

Table 3. Hypervisor Meta-Data

| VM meta-data | Description |
|---|---|
| VM OS | Ubuntu 12.09 |
| VM name | Ubuntu 1 |
| VM IP | 10.245.240.64 |
| VM Port | 27000 |
| VM MAC address (HWaddr) | C2:0C:30:C0:00:44 |
| VM UUID | 20972b8f-f33a-4e4c-bff8-2283aoe1a2b3 |
| Allowed operation | vm-migrate, provision, vm-start, vm-resume |

Table 4. Meta-Data

| Snapshot meta-data | Description |
|---|---|
| Memory overhead | 12582912 |
| name | snap8 |
| VCPU-max | 2 |

Table 5. Snapshot Meta-Data

***2) Snapshot content and deleted files:*** After retrieving the meta-data successfully, the next challenge is whether the investigator can extract suspect files from the .xva file without the need to transfer or rebuild the virtual environment. Using the DFF tool, the snapshot contents such as text, multimedia, and emails, can be retrieved and examined, as depicted in Figure 8. Furthermore, in the case where the organization requests a particular file format that is not specified in the tool, the investigators can add the header of the desired file format and run the tool to examine the snapshot (as is the case with most of the file carving tools).

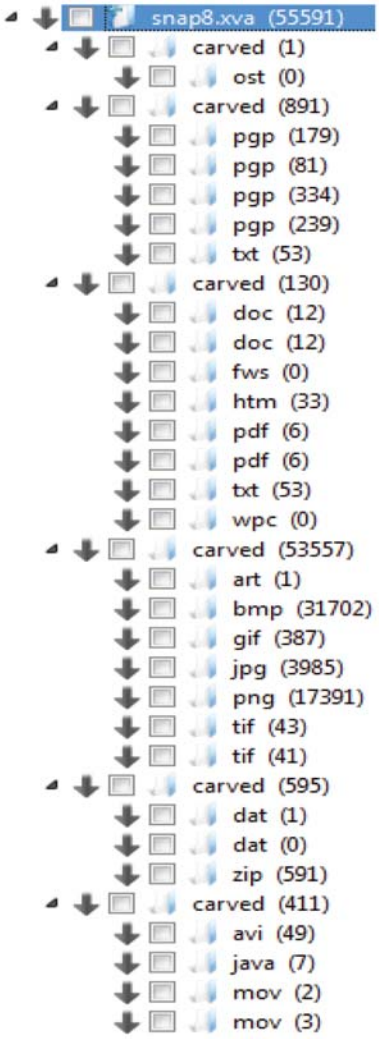


Figure 8. List of Retrieved Files from the Snapshot

Besides previous findings, the snapshot file holds artefacts of deleted files. A further experiment was conducted to examine the impact of different file deletion techniques, and several tests have been conducted.

We considered two delete actions that might be performed by the user, either using the "delete" (mark the file as deleted), or "shred" (divide the file into smaller files and perform repeated overwrite on its content).

| Delete Options | Snapshot State |
|---|---|
| Using OS interface | Retrieved successfully |
| Using shred command | Retrieved successfully |

Table 6. The State of A Rtefact With Different Delete Methods

Since XenServer provides COW snapshots, the system copied the file before writing to it. This is considered to be one of the "shred" command limitations. Hence, the evidence was retrieved successfully.

**10.5 Analysis**

Arranging events chronologically is a good way of telling a clear, concise story. The existing tools were not able to generate a timeline of suspect actions, even though, each file had associated modified, access and create (MAC) time and was successfully detected by the DFF tool. Hence, a possible abstract algorithm will aid to generate a timeline of snapshot content as described in Algorithm 1.

**Algorithm 1** Generat Timeline

1: **while** *!EoF* **do**

2: *searchP erFileExtension();*

3: **if** *Extension Found* **then**

4: *modifiedList:Add(File;Mtime);*

5: *accessedList:Add(File; Atime);*

6: *createList:Add(File;Ctime);*

7: **end if**

8: **end while**

9: *SortAscending(modifiedList);*

10: *SortAscending(accessedList);*

11: *SortAscending(createList);*

Algorithm 1 will generate a sorted list of modification, access, and creation actions as performed by the user.

**10.6 Reporting**

After a successful and accurate examination and analysis of suspected VM snapshot, a final report that includes all the findings can be provided to the organization.

**11. Results And Discussion**

The practical assessment proved that the existing snapshot process (COW) is a good candidate to produce forensically sound evidence regardless of its format and without the need to recreate the virtual environment. Therefore, we map the technical requirements to the COW method as follows:

• The integrity of the snapshot file can be verified.

• The logical acquisition of the snapshots results are shown in Table 7.

| Evidences | Physical Storage | VM Snapshot |
|---|---|---|
| Existing files | Yes | Yes |
| Deleted files | Yes | Yes |
| Unallocated space | Yes | Yes |
| Slack space | Yes | No |

Table 7. The Examinaiton of Snapshot In Comparison To The Physical Storage

The XenServer may include the slack space within the

desired VM snapshot by importing *Slackware* repository. However, this will tremendously increase the size of the snapshot and may affect the performance of the system.

• The examiners can extract the evidence independently from the original storage.

• The target storage is wiped, where XenServer claims that by taking a snapshot, a clean copy of the Virtual Desk Image is created.

• Zero snapshot latency, using both the management console and the CLI, the snapshot was taken instantly.

As a result, we assess the ability of the proposed DF-CBSS in examining the snapshot file. Therefore, we conclude that the snapshot examination result is forensically equivalent to the physical storage examination.

## 12. Conclusion And Future Work

In this paper, we have investigated the possibility of examining VM snapshots storage dumps. Previous studies proposed several digital forensic requirements that need to be satisfied by the current snapshot method. For a particular case investigation, re-creating the virtual environment is a costly and time-consuming process. The main point that we are emphasizing on is to utilize the amount of data that can be acquired from the snapshot dumps.

We assessed the feasibility of storage acquisitions using existing digital forensic tools. Four categories of tools had been used; (1) computer forensic, (2) memory forensic, (3) file carver, and (4) binary editor. We showed that the existing tools were capable of retrieving under investigation files (text and image files), however, further research needed to investigate the possibility of recovering encryption keys and passwords.

As a future work, further study will be conducted on the big data distributed storage snapshot where the HaDoop framework will be used as a test platform. The main challenge in a HaDoop system is the data storage methodology, which is managed by the HaDoop Distributed File System (HDFS), where the HDFS is designed to divide storage into chunks of fixed sizes and store them on data nodes, however, their meta-data is stored on a master node. Using an Online Image Viewer, forensic examiners can examine the meta-data in a human readable format [2], however, examining storage snapshot as a binary dump needs to be studied further.

## References


[1] Almulla, S., Iraqi, Y., Jones, A. (2014). A state-of-the-art review of cloud forensics, *Journal of Digital Forensics, Security and Law*, 9 (40) 7–28.

[2] Almulla, S., Iraqi, Y., Jones, A. (2013). A distributed snapshot framework for evidence extraction and event reconstruction from cloud environment, *In:* IEEE International Conference on Cloud Computing Technology and Science (CloudCom), p. 699–704.

[3] Almulla, S., Iraqi, Y., Jones, A. (2013). Cloud forensics: A research perspective, *In:* 9th International Conference on Innovations in Information Technology (IIT), p. 66–71.

[4] Cisco System (2015). Virtualization at small and medium sized firms on the rise, http://www.ciscomcon.com/sw/themes/12949/site images/Cisco-WhitePaper-Virtualizatio-FINAL.pdf, (2013), Accessed 22-12- 2015.

[5] Dykstra J., Sherman, A. (2013). Design and implementation of FROST: digital forensic tools for the OpenStack cloud computing platform, *Digital Investigation*, V. 10, p. 87–95.

[6] Huseinovic A., Ribic, S. (2013). Virtual machine memory forensics, *In:* 21st Telecommunications Forum (TELFOR), p. 940–942.

[7] Sh. Zhang, Wang, L., Han, X. A kvm virtual machine memory forensics method based on vmcs, *In:* Computational Intelligence and Security (CIS), 2014 Tenth International Conference on, 2014, p. 657– 661.

[8] Guangqi, L., Lianhai, W., Shuhui, Z., Shujiang, X., Lei, Z. (2014). Memory dump and forensic analysis based on virtual machine, *In*: Mechatronics and Automation (ICMA), 2014 IEEE International Conference on, 2014, p. 1773–1777.

[9] Saibharath, S., Geethakumari, G. (2014). Design and implementation of a forensic framework for cloud in openstack cloud platform, *In*: Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on, 2014, p. 645–650.

[10] Naik, N., Kumar, K., Vasumathi, D. (2014). Securing information by performing forensic and network analysis on hosted virtualization, *In*: 2014 International Conference on Computer and Communications Technologies (ICCCT), p. 1–7.

[11] Banas, M. (2015). Cloud forensic framework for iaas with support for volatile memory, M.S. thesis, School of Computing, National College of Ireland, Dublin, Ireland.

[12] Diane, B., Gregory, K., Eds., (2010). Virtualization and Forensics: A Digital Forensic Investigator's Guide to Virtual Environments, Elsevier.

[13] Garimella, N. (2006). Understanding and exploiting snapshot technology for data protection, http://searchstorage.techtarget.com/definition/ storage-snapshot, (2006), Accessed 10-07-2013.

[14] Inc. Citrix Systems (2009). Citrix xenserver: Understanding snapshots, http://www.scribd.com/doc/69565643/ XenServer-Understanding-Snapshots-v1-1, (2009), Accessed 21- 07-2013.

[15] Stephenson, P. (2003). Putting the horse back in front of the cart, *In*: Third Digital Forensics Research Workshop, 2003.

[16] FTK. Forensics tool kit (FTK) computer forensics software, http:// accessdata.com/products/computer-forensics/ftk, Accessed 25-12-2016.

[17] Narzisi, G. (2007). Scalpel, http://scalpel.sourceforge net, Accessed 25-02-2017.

[18] Mh-nexus, (2007). Freeware hex editor and disk editor, https://mh-nexus.de/ en/hxd/, Accessed 25-02-2017.

[19] Altheide., C, Carvey, H. (2011). Digital Forensics with Open Source Tools, Syngress Publishing, 1st edition.

[20] NIST, (2016). Guide to Integrating Forensics Techniques into Incident Response, http://csrc.nist.gov/publications/nistpubs/800-86/SP800-86.pdf, 2006, Accessed 25-12-2016.

[21] Palme, G. (2001). Report from the first digital forensic research workshop (DFRWS), Tech. Rep., Air Force Research Laboratory, Rome Research Site, 2001.