Chapter 5

ANALYZING THE IMPACT OF A VIRTUAL MACHINE ON A HOST MACHINE

Greg Dorn, Chris Marberry, Scott Conrad and Philip Craiger

Abstract As virtualization becomes more prevalent in the enterprise and in personal computing, there is a great need to understand the technology as well as its ramifications for recovering digital evidence. This paper focuses on trace evidence related to the installation and execution of virtual machines (VMs) on a host machine. It provides useful information regarding the types and locations of files installed by VM applications, the processes created by running VMs and the structure and identity of VMs, ancillary files and associated artifacts.

Keywords: Virtualization, virtual machine, VMware, Parallels

1. Introduction

The Sarbanes-Oxley Act of 2002, which was passed as a result of major accounting scandals, requires corporations to have the digital forensic capability to identify and recover information and documentation related to their business operations. As virtualization becomes more prevalent, there is a great need to understand the technology and its impact on evidence recovery.

This paper investigates several key aspects related to the behavior of virtual machines (VMs) and the impact of VMs on host systems. The structure of VMs is explored by experimenting with two popular virtualization systems, VMware [14] and Parallels [11]. The associated VMs are analyzed to identify the artifacts produced during VM creation and deletion and the impact of their processes on the host machine. Also, VM behavior is investigated by deleting individual files from within a VM and observing the results. Commercial off-the-shelf forensic software

systems, including EnCase [6] and Forensics Toolkit [1], are used to locate artifacts that remain on the host system during and after the use of virtualization software and to determine if the artifacts are recognizable as belonging to a VM or the host machine.

2. Testing Methodology

Our method for researching and testing the behavior of virtualization software and VMs involved four steps: (i) virtualization software was installed on a new desktop computer and used to create several VMs; (ii) test files were placed on the VMs, executed and then deleted, and the changes in the VMs and host system were recorded; (iii) the hard drive on the host machine was imaged; and (iv) the image was analyzed using forensic software.

2.1 Host Machine Configuration

A new Dell Optiplex 755 machine (referred to as the host machine) was used for the tests. The drive was wiped using a Knoppix Live CD version 5.1.1 [9] and the command `dd if=/dev/zero of=/dev/sda`.

Next, Windows XP SP3 was installed, followed by VMware Workstation 6.0.3 [15] and Parallels Workstation 2.2 [12].

2.2 Virtual Machine Configuration

VMware Workstation was used to create two VMs: Windows XP SP3 and Ubuntu 8.04 Desktop [3]. The VMs were created with the default settings within VMware using the "Typical" option. The guest operating system type was chosen from the drop down list, the VM was assigned the default name provided by VMware, bridged networking was assigned as the network connection, and the size of the virtual hard disk was set to the default 8.0 GB.

The Parallels Workstation was also used to create two VMs: Windows XP SP3 and Ubuntu 8.04 Desktop. The VMs were created with the default settings within Parallels using the "Create a Typical VM" option. The guest operating system was chosen using the appropriate option, the VM name set to the default provided by Parallels, a new folder was created by Parallels for the newly created VM files, a 32 GB virtual hard disk was created and bridged networking was used. The guest operating systems were installed using installation CDs or DVDs for each VM in an identical fashion to the installation of the operating system on a physical machine, where the physical machine boots from the installation media and the operating system is installed.

Several types of files were used to test VM behavior: `.mp3`, `.jpg`, `.htm`, `.txt`, `.xls` and `.pdf`. The files were copied to the host machine and to all four VMs via a network shared drive and placed in folders labeled Test Files. The test files for all the Windows machines (including the host machine) were placed on the Desktop and in the My Documents and WINDOWS folders. For the Ubuntu VMs, the test files were placed on the Desktop and in the Home folder. The test files were given different names to identify them by their type, operating system and location. Next, one file was deleted from each location on all the VMs. Then, one VM was deleted from each type of virtualization software, Ubuntu from VMware and Windows XP from Parallels. This was done in order to compare and contrast the behavior of the two types of operating systems across the two virtualization software systems.

2.3 Analysis Machine Configuration

An established Dell Optiplex 745 machine with Microsoft Windows XP SP2 was used as the analysis machine. This machine contained EnCase versions 5.5.11.0 and 6.10.0.25, Forensics Toolkit 1.70.1 and Xways Forensics 14.2 [19]. The machine also contained WinHex [18], Hex Editor Neo [8] and TextPad [7] for viewing and altering files.

2.4 Host Machine Imaging

The hard drive on the host machine was imaged immediately after every major change to the file system. The drive was imaged after the initial configuration was complete, including the deletion of the specified VMs and test files. A second image was created after clearing the Recycle Bin/Trash on the host machine and all the VMs. A third hard drive image was created after the Disk Cleanup tool was used on the virtual drive of the VMware XP VM; the `sudo apt-get clean` command was executed on the Parallels Ubuntu VM; and the Disk Cleanup and defragmentation tools were used on the host machine.

A special protocol was used for imaging drives. First, the host machine was shut down and its hard drive removed. Next, the removed hard drive was connected to the analysis machine via an UltraBlock SATA Bridge Write Blocker [5]. The image was then acquired using EnCase and saved to a different case file labeled according to the specific state of the drive at the time it was imaged (e.g., Case 10.1 Deleted Files Image).

3. Virtual Machine Structure

The structure of a VM depends on the type of virtualization software used to create it. Specific artifacts and evidence created by virtualization

Table 1. Virtual machine file types.

VMware	
`.vmdk`	Virtual hard disk file
`.vmx`	Virtual machine configuration file
`.vmxf`	Supplemental configuration file
`.nvram`	BIOS for virtual machine
`.vmsd`	Dictionary for snapshots
`.log`	Virtual machine activity log
Parallels	
`.hdd`	Virtual hard disk file
`.pvs`	Virtual machine configuration file

software and VMs include (but are not limited to) registry entries, VM files, processes and virtualized hardware. This information pertains to the standard objects related to VMs and virtualization software and is not exhaustive.

3.1 File System

VMware and Parallels create VMs in a similar manner. A main folder is created for the type of virtualization software, a subfolder for each VM is created in the main folder, and the files comprising the VM are located in that subfolder.

Each folder contains a subfolder for the VM, which is created using a naming scheme to reflect the operating system used by the VM (the default naming scheme). The individual VM folders contain a number of files that comprise the VM. VMware creates a default set of six files [16] while Parallels creates two files (Table 1). Of these files, the most important are the virtual hard disk files (`.vmdk` and `.hdd`) and the configuration files (`.vmx` and `.pvs`).

3.2 Registry

The host machine on which VMs execute contains several registry entries related to virtualization. These entries, shown in Figure 1, define the types of virtualization software and/or the files that relate to the virtualization software used to create the VMs and the VM configuration files [10].

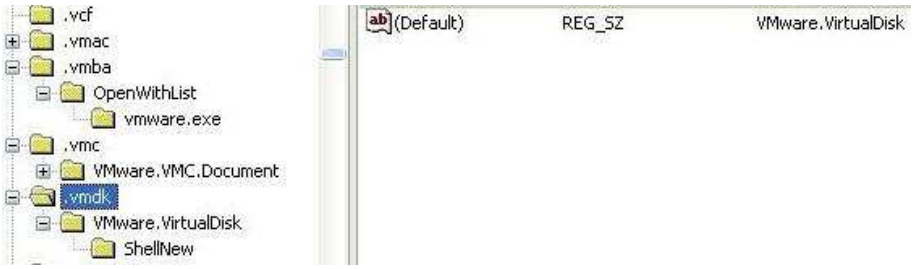

Figure 1. `HKEY_CLASSES_ROOT` registry entry.

The key shown in Figure 1 was found using the "Find" option in the registry using the value "vmware." Additional searches for the VM files provided the locations of their keys.

3.3 Virtualization Software

One of the most obvious artifacts of virtualization on a host machine is the presence of virtualization software. The two most popular systems of this genre are VMware and Parallels. Open source virtualization software systems include Qemu [2], Xen [4] and VirtualBox [13]; however, these systems are not considered in this work.

Figure 2. Program Files entries.

Most types of software installed on a machine can be identified based on the naming conventions used for their file entries in the host machine file system; these file entries are generally located in the Program Files folder of the host system (Figure 2). Beyond the standard file system entries, additional evidence of virtualization software may persist in the WINDOWS Prefetch folder and the WINDOWS Temp folder. These entries can remain on the host machine even after VMs and the virtualization software are deleted.

3.4 Virtualized Hardware

VMs, like physical machines, rely on hardware to facilitate network connections and to read/write different types of media. To that end, virtualized hardware is created for each VM type (VMware and Parallels).

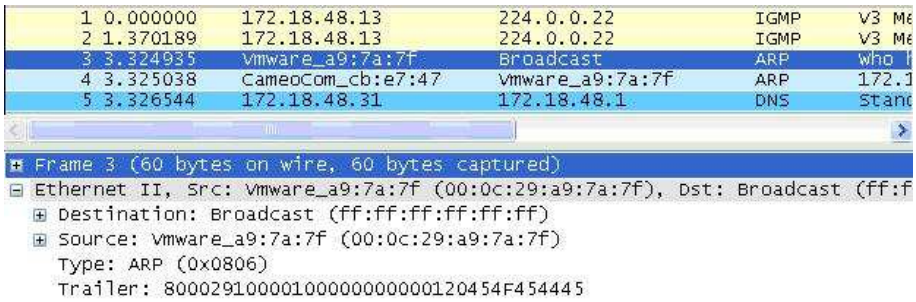

Figure 3. Wireshark capture of VM traffic.

Virtualized hardware may include a virtual hard drive, virtual display, virtual network device/adapter, virtual USB interface/manager, virtual SCSI/RAID controller and mouse/pointing device.

Evidence of the use of virtualized hardware can be found as well. The virtualized network interface card for a VM identifies itself as a VMware product during a Wireshark [17] capture. Note that the MAC address of the card is also recorded (Figure 3). VMware signatures are indeed pervasive on all types of virtualization hardware.

Figure 4. Process entries.

3.5 Processes

Additional traces of virtualization software can be located within running processes and the startup configuration of a host machine. Figure 4 shows how these areas within a host machine list the virtualization software that is running on the machine.

VMware and Parallels maintain process items on a host machine whether or not a VM is running. When one these programs is running, the number of processes associated with it increases as when any new program or application executes on a machine.

Table 2. Suspended state and snapshot files.

VMware	
`.lck`	Lock file created when VM is running
`.vmss`	Suspended state file
`-snapshot.vmsn`	Snapshot file for VM reversion
`-snapshot.vmem`	Snapshot of VM memory
Parallels	
`.sav`	Suspended state file

4. Suspended State and Snapshot Artifacts

Beyond the standard set of files created with each new VM, several types of files are created when a VM is suspended or its current state is saved as a backup (snapshot). A key feature related to the suspended state and snapshot files is the listing of configuration settings that detail VM attributes, i.e., the operating system installed on the VM, virtualized hardware attached to the VM, and the paths to any ISOs that were used and hard disk files that were accessed. Table 2 provides a listing of these files.

An examination of the `.vmss` and `-snapshot.vmsn` files indicates that they contain information that is nearly identical to the configuration (`.vmx`) file of a VM, i.e., they contain VM settings. These files can be an important source of information when analyzing a drive from the host machine or VM with regard to the existence of a VM or the behavior of a running VM.

The suspended state files (`.vmss` and `.sav`) are not retained when a VM is restarted. The files are deleted in the same manner as the entire VM – they are not sent to the Recycle Bin and are simply flagged as being available for overwriting. These files can be recovered using the methods described below.

Snapshot files are maintained in the VM folder until they are either deleted individually or deleted as part of the VM. In both cases, the files are sent to the Recycle Bin or Trash. As with the suspended state files and other files associated with VMs, these files are recoverable from the host machine hard drive.

A snapshot function is not available for Parallels Workstation for Windows. However, Parallels for Macintosh does support a snapshot function and creates four different files (`sav`, `.pvc`, `.mem` and `.png`). These

files are similar to their VMware counterparts in that they contain data about the current state of the VM.

5. Deleting and Identifying Virtual Machines

A VM can be deleted by deleting it from the virtualization software or by deleting from the host machine the individual files and folders that comprise the VM. Deleting a VM from a host machine using the virtualization software simply places the files back in unallocated space. The files are not put into the Recycle Bin (like most other files) because they are typically larger than what the Recycle Bin would allow. Some files and folders from the host machine could be put in the Recycle Bin if they are small enough; however, the hard disk file is typically too large and is, therefore, placed in unallocated space. These files are intact (and recoverable) until they are overwritten.

EnCase was used to view and analyze the host machine hard disk images. The contents of the hard disks were viewable and searchable and the locations for VMware VMs were explored. The two VMs created with VMware were visible, the Windows XP VM was intact and the Ubuntu VM was flagged as being deleted. A search of the VM parent folders revealed files that comprised a VM created with VMware; the files were intact and were viewable with EnCase.

5.1 Identifying Deleted Virtual Machines

The hard disk file associated with a VM is important because it contains all the information about the VM. The principal difference between the structure of the VM hard disk file and that of the host machine hard disk is the information written at the beginning of the disk specific to the types of headers and other configuration data needed by the VM. Once the sectors containing the operating system are reached, the VM and the host machine disks appear identical. We discovered that the VMware virtual hard disk files begin with the header `KDMV` and Parallels hard disk files begin with the header `WithoutFreeSpace`. Using these values, hard disk files can be searched for in the unallocated space on the host machine drive.

5.2 Recovering Deleted Virtual Machines

During the course of an investigation, it may be necessary to extract a deleted VM from the unallocated space of the hard disk. Searches of the unallocated space using known headers yielded the locations of deleted VMs. Once the hard disk file is located, the file can be treated as an actual hard disk with respect to locating the Master Boot Record

Figure 5. EnCase view of added VM hard disk file.

within the file. Our research showed that the partition table was typically located at the relative offset `001101ca` within the hard disk file for VMware and at the relative offset `003f81da` for Parallels. This information allowed us to locate the type and size of the partitions for extraction purposes. VMware typically breaks the virtual hard disk into 2 GB portions until the virtual hard disk size specified by the user is reached; this enables a size value to be used when attempting to extract a VM created by VMware. Once the size of the hard disk file is known, EnCase can be used to extract the file and save it to the analysis machine.

The virtual disk of a VM can be exported to the analysis machine using EnCase and then added back to an open case in EnCase as a new device, rendering the single `.vmdk` file as if it were another complete hard drive containing an operating system. An example is shown in Figure 5.

At this point, the VM hard disk can be searched for relevant information or files as in any forensic investigation. While other files are necessary to run the VM in the virtualization software, the hard disk file is the most important aspect of a VM.

6. Deleted Files on a Virtual Machine

It is important to understand how files are treated within a VM. The behavior of a VM hard disk file is identical to that of a physical hard drive on a physical machine. The operating system is installed in the same manner on both types of machines, ensuring that applications and files are handled in a similar manner.

We created a set of test files to identify what happens when a file in an VM is deleted and then the VM itself is deleted. A file deleted in a VM is treated in the same way as one deleted in a physical machine: the file is moved to the Recycle Bin or Trash, where the first bit of the file is flagged to indicate that it can be overwritten as necessary. Recovering a deleted file from a VM is similar to recovering one from a physical machine. Depending on the operating system, this can range

from simply restoring the file from the Recycle Bin or Trash to using a third-party application to restore the file.

In the case of a deleted VM, it is necessary to locate and then extract the VM hard disk file in the same manner as a physical drive; this provides access to all the data (intact and deleted) that resides on it. Further analysis of the VM files can be performed by recovering the hard disk file and viewing it with EnCase as described above or by recreating the entire VM as described below.

7. Recreating a Virtual Machine

Standard forensic analysis focuses on the hard drive of a target machine and possibly on the additional hardware installed on the machine. Virtualization changes this perspective because of the introduction of virtual software and hardware. It is possible – and sometimes necessary – to recreate intact and deleted VMs on a host machine.

In our experiments, the VMware Workstation Windows VM was found to be intact in the file system of the host machine. Once the appropriate files were located for the VM, the `Copy/Unerase` function was used to copy them to a folder called RecoveredWindows on the analysis machine. As each file was copied, it was renamed to `recovered(VMware file type)` (e.g., `recoveredWindows XP Professional.vmdk`). Then, the newly created folder was opened and the configuration (`.vmx`) file was opened, which automatically opened VMware Workstation. The newly created VM was started and a message appeared to indicate that the VM had been copied or moved. The newly renamed `.vmdk` file path had to be provided for the VM to start properly. Using the default settings, the VM started successfully and all the data was intact as it originally was on the host machine. The test files installed on the VM were intact and the `md5sum` hashes of the files matched the original test file values.

We encountered two scenarios with respect to deleted VMs. In one instance, the VMware Workstation Ubuntu VM was deleted from the host machine and was partially overwritten by another program (AVG anti-virus log file). In the second instance, another image of the host machine showed the VMware Workstation Ubuntu VM as simply being deleted with no overwriting.

In the first instance of deleting the Ubuntu VM, the entire folder for the VM was still listed in its original location in the host machine file system with all the VM files. The `.vmdk` file was indicated as deleted and was, in fact, overwritten. The `Ubuntu.vmdk` file was copied using the `Copy/UnErase` function as described above. Although the file was overwritten, it was determined (upon searching the data using a hex viewer)

that much of the original data still persisted in the file; the overwriting was confined to the beginning of the hard disk file and did not corrupt any operating system information. Since the virtualization hardware is standardized for a given version of VM software (e.g., VMware Workstation 6), recovering partially overwritten data is a possibility. Depending on the position of the lost data in the VM configuration file, it is possible to insert the standardized values to fully recover the configuration file and allow the VM to function. The recovered Ubuntu `.vmdk` file was opened using Hex Editor Neo. A VM from the analysis machine, Ubuntu `Desktop.vmdk`, was opened using the same hex editor. The beginning of the Desktop version file, equal to the portion that was determined to be overwritten, was copied and then pasted into the recovered file. This was saved as `ubunturepaired.vmdk`.

A similar methodology was used to recover and correct the VM configuration file for the deleted Ubuntu VM. The original file was copied from the image via `Copy/UnErase` and renamed as `ubunturepaired.vmx` to reflect the naming scheme of the recovered VM. Unnecessary data written to the file (as compared with an intact configuration file) was deleted and the entries were modified to reflect `ubunturepaired.vmdk`, the new name of the VM virtual hard disk file. This configuration file was renamed as `ubunturepaired2.vmx`.

Opening the recovered `ubunturepaired2.vmx` file started the VMware Workstation software. As with the recovered Windows VM, a message was displayed concerning the moving or copying of the VM. Once this message was acknowledged, the recovered Ubuntu VM started and ran normally. All test files were intact and the metadata was visible and correct.

The second instance in which the Ubuntu VM was deleted displayed no traces of the VM in the host machine file system. Searches were conducted to ascertain the location of the `.vmdk` file. As described above, the file was located and extracted to its own folder on the analysis machine. This was the first step in attempting to recreate the deleted VM. Next, VMware Workstation was run on the analysis machine and a new custom VM was selected for creation. The only difference in creating the VM for recovery was the selection of an existing hard drive instead of creating a new hard drive for the VM. Once this was done, the VM started normally and all the test files were intact. No additional changes were necessary for the VM to run as it was originally configured.

The next step in the recovery process for the completely deleted VM was to extract the configuration file (`.vmx`). This file was located and extracted to the same folder in an attempt to recreate the conditions under which the original VM was created. The process of recreating the

VM from the second host machine image was not immediately successful using the two extracted files from the unallocated space. Similar to the recreation of the deleted Ubuntu VM, entries were modified in the configuration file to enable it to run the VM. Specifically, the path to the recovered `.vmdk` file was modified and other extraneous characters copied during the extraction process were removed in order for the configuration file to match the comparison file from the analysis machine.

When the configuration file was opened, VMware Workstation started automatically. The message regarding the moving or copying of the VM was displayed and was acknowledged as being copied. The VM started correctly and all the information was present and intact.

Further research into the process of extracting deleted VMs from the unallocated space on the host machine revealed no discernible differences between the VMs recreated with only the `.vmdk` file versus those recreated with the `.vmdk` and `.vmx` files as well as any other standard file normally created for a new VM. When the extracted/recreated VM was started on the analysis machine, new files were created to complete the file set for the VM, which negated the effects of the other files that were extracted and placed in the VM folder.

8. Conclusions

The recognition and understanding of the role that virtualization software and VMs play in computing environments in important in digital forensic investigations. Investigators and examiners must be aware that artifacts are produced during the creation and execution of VMs and that many artifacts are recoverable even after the VMs are deleted.

Our research has shown that virtualization software and VMs produce numerous artifacts in a host machine. Specific behavior patterns, such as deleting certain types of virtualization files and even entire VMs, can be used to narrow searches for artifacts. Furthermore, the ability to identify, extract and recreate entire VMs is very useful when investigators and examiners have to determine the circumstances under which the VMs were used.

References

[1] AccessData Corporation, Forensic Toolkit 1.7, Linden, Utah (www.accessdata.com).

[2] F. Bellard, Qemu (bellard.org/qemu).

[3] Canonical, Ubuntu 8.04, London, United Kingdom (www.ubuntu.com).

[4] Citrix Systems, What is Xen? Fort Lauderdale, Florida (www.xen.org).

[5] Digital Intelligence, UltraBlock SATA Bridge Write Blocker, New Berlin, Wisconsin (digitalintelligence.com).

[6] Guidance Software, EnCase 5 and 6, Pasadena, California (guidancesoftware.com).

[7] Helios Software Solutions, TextPad, Longridge, United Kingdom (www.textpad.com/index.html).

[8] HHD Software, Free Hex Editor Neo, London, United Kingdom (www.hhdsoftware.com/Products/home/hex-editor-free.html).

[9] Knopper.Net, Knoppix Live Linux Filesystem, Knoppix 5.1.1 Release, Schmalenberg, Germany (www.knopper.net/knoppix/index-en.html).

[10] T. Liston and E. Skoudis, On the cutting edge: Thwarting virtual machine detection (handlers.sans.org/tliston/ThwartingVMDetection_Liston_Skoudis.pdf), 2006.

[11] Parallels, Parallels Optimized Computing, Neuhausen am Rheinfall, Switzerland (www.parallels.com).

[12] Parallels, Parallels Workstation 2, Neuhausen am Rheinfall, Switzerland (www.parallels.com/en/products/workstation).

[13] Sun Microsystems, VirtualBox, Santa Clara, California (www.virtualbox.org).

[14] VMware, VMware, Palo Alto, California (www.vmware.com).

[15] VMware, VMware Workstation 6, Palo Alto, California (www.vmware.com/products/ws).

[16] VMware, What files make up a virtual machine? Palo Alto, California (www.vmware.com/support/ws5/doc/ws_learning_files_in_a_vm.html).

[17] Wireshark Foundation, Wireshark, San Jose, California (www.wireshark.org).

[18] X-Ways Software Technology, WinHex, Cologne, Germany (x-ways.net/winhex/index-m.html).

[19] X-Ways Software Technology, X-Ways 14.2, Cologne, Germany (x-ways.net/forensics/index-m.html).