FORENSIC RECOVERY OF EVIDENCE FROM DELETED ORACLE VIRTUALBOX VIRTUAL MACHINES

by

Cherilyn Neal

A Capstone Project Submitted to the Faculty of

Utica College

December 2013

in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Cybersecurity

Abstract

The increased use of virtualization among government agencies, private enterprise, educational institutions and private users has opened a new avenue of research in the field of digital forensics. Virtual machines are being used for testing of software applications, malware research, technology education, and the expansion of an organization's infrastructure while reducing costs. The benefits of virtualization include the ability to run several virtual machines with several different operating systems on a single hardware platform, as well as keep the guest and host environments isolated. This isolation is predicated on the way the virtualization technology is designed, making it an attractive solution as a testing environment. This isolation also makes it an attractive solution for use in cybercrime. The ease by which a virtual machine can be deleted or reverted to a previously clean, saved state classifies as an anti-forensics technique; obfuscating or destroying digital evidence that may have been beneficial in the investigation of a crime. Techniques have been developed to acquire virtual machine images for analysis and several virtual forensic platforms that are pre-built with forensic tools are freely available, but very little research has been dedicated to the problem of recovering evidence of activity after a VM has been deleted or reverted. This research sought to ascertain whether any evidence of the activity generated inside a virtual machine could be recovered, as well as the ability to restore recovered virtual machine files to a functioning virtual machine that could then be examined. Keywords: Cybersecurity, Professor Cynthia Gonnella, forensics, virtual machine, Oracle VirtualBox, cybercrime, anti-forensics.

Acknowledgements

I would like to thank the incredibly supportive faculty and staff at Utica College for facilitating a smooth transition from my undergraduate program to my graduate program. This program was more than challenging, and I appreciate the effort and experience that all my professors brought to the curriculum. I wish to express my gratitude to Professor Joseph Giordano, Professor Anthony Martino, Professor Randall Nichols, and Professor Vernon McCandlish for their encouragement and assistance. I would especially like to thank my thesis advisor Professor Cynthia Gonnella and my second reader Kim Smathers for all their work in helping me to take this project from an exciting possibility to a finished product. This Master of Science in Cybersecurity Capstone is dedicated to my husband Toby and my daughter Arianna. I appreciate their support and understanding through all of the family nights and weekends that they sacrificed for my long-held goal to graduate with my Master's degree from Utica College. I could not have completed this project without their patience and support.

Table of Contents

List of Illustrative Material

Forensic Recovery of Evidence from Deleted Oracle VirtualBox Virtual Machines

The purpose of this research was to examine the possibility of recovering forensic evidence of user activity within an Oracle VirtualBox virtual machine (VM) that the user deleted or reverted to a restored point. This body of research investigated the potential availability of forensic artifacts left on the host drive after an Oracle VirtualBox VM is deleted or rolled-back to a snapshot, a feature of VirtualBox that allows the user to create a saved state of the VM (Wallen, 2013). The user can then roll back to a previously "good" running state in the event that the VM becomes corrupted, to revert to a clean pre-test state, or to discard changes or evidence. The purpose of the analysis was to examine acquired hard drive image files and volatile data from a controlled experimental environment for any overflow of user activity data from the VirtualBox guest operating system (OS) to the host OS. The analysis also examined whether or not an Oracle VirtualBox VM could be reconstructed from a recovered or partially recovered virtual disk image file. The research also considered the cost benefit of allocating the necessary resources to perform the associated tasks when strategizing a plan for forensic analysis.

Oracle VirtualBox

Oracle VirtualBox is a cross-platform, open source virtualization software that allows one computer to run multiple OSs in a simulated environment (Oracle Corporation, 2013). As an open source product, the source code is available for the user to review and modify if they wish. The VirtualBox base package is free to download and use for personal use with no limit to distribution (Oracle Corporation, 2013). The open source nature of the product provides a flexible and economical solution for testing software, malware research, duplicating environments and educational purposes (Shavers, 2008). As is common with many beneficial tools, cybercrime activities can be facilitated by using VMs as disposable OSs for discarding

evidence of criminal activity (Shavers, 2008). Virtual machines offer cyber criminals an environment that is portable and easily destroyed by deleting the VirtualBox files or restoring to a previously saved snapshot, leaving seemingly no trace of their activity behind for forensic analysts (Shavers, 2008).

Although ample research has been conducted on the use of VMs as forensic platforms and the forensic methods of collecting evidence from recovered VMs, little information was found during the course of researching the subject through web and library resources that addressed the recovery of evidence from deleted or reverted VMs. This paper explains virtualization technology, the benefits and limitations of the technology, a review of the existing work, and an analysis of acquired memory and hard disk image in an effort to answer the research questions. Digital forensic analysts can benefit from this research as adding another vector for the investigation of cybercrime.

The Benefits of Virtualization

Virtualization has become a popular solution for organizations of all sizes including government agencies, private enterprise, educational institutions, and private users because of the flexibility and scalability that it offers (Hirwani, Pan, Stackpole, & Johnson, 2012). In 2009, Gartner reported that 18% of server workloads were running on virtualized servers and that the number would grow to almost 50% by 2012 (Messmer, 2009). Virtualization allows one or more "guest" OSs to run on top of a "host" OS (Liston & Skoudis, 2006). Each guest OS runs in an emulated virtual machine environment (VME), managed by a hypervisor or virtual machine monitor (VMM) (Barrett, 2010). The hypervisor allocates the necessary physical resources including CPU, memory, network and storage and manages the communications between VMs and these physical resources (Bazargan, Yeun, & Zemerly, 2012). The VMM allows the guest

OS to access virtual and real hardware in order to function, but the guest is a self-contained virtual environment that functions like a separate OS (Liston & Skoudis, 2006).

The guest OS can be run in the VME while maintaining the integrity of the host OS, allowing one server or computer to run several OSs at once. The benefit of this solution is a reduction in costs for organizations in hardware and space, as one server can host multiple guest VMEs (Liston & Skoudis, 2006). The virtual environment can be shut down and the user is returned to the host OS. Files can be shared between host and guest through a shared network if desired, or the guest system can be isolated from the host system and network (Shavers, 2008). In theory, the guest and host OSs are isolated from one another, which is why virtual environments are commonly used for testing of software applications and for malware research (Shavers, 2008). Educational institutions use virtualization technology to teach students how to use different OSs; the installation of several OSs on a single desktop or laptop PC saves time and resources (Shavers, 2008). The individual user can apply virtualization to test software, new OSs or patches and updates. Malware researchers use virtualization to test malware in a controlled environment while protecting the production environment from infection and to observe behavior in different environments (Shavers, 2008). These features are driving the increased adoption of virtualization. As is true with all technological solutions, virtualization has limitations that must be weighed before implementation.

Limitations of Virtualization

Virtualization can negatively affect performance; the CPU processing power needed to run virtualization software can be considerable (Bazargan, Yeun, & Zemerly, 2012). The RAM allotted to the VM will not be available for the host machine while the VM is running; therefore, the VM should be allotted enough RAM to perform while retaining enough RAM for the host

system to perform. Oracle suggests allotting RAM so that there is at least 256 to 512 MB left for the host system (Oracle Corporation, 2013). Each installed VM that is running will require its allotted RAM, and the performance of the host system can become sluggish with so few resources left to function.

While the VMM manages the emulation of hardware, not all hardware platforms are supported by virtualization (Apriorit Inc, 2011). Depending on what the requirements are and what the VM is designated for, this may or may not be an impediment. For most use, the natively supported hardware is enough. Hardware resources are also a concern; mainly the hard drive space required to run multiple VMs. Each VM image takes up a great deal of space depending on how much it has been allotted, and the VM images can quickly fill a hard drive.

Antivirus software installed on the host system is not available for use by the guest system; a separate antivirus product must be installed on the VM to protect it. This is true for all other software applications installed on the host. Productivity, games, and other required or desired software applications must also be installed on the VM. These limitations may not create a problem for the majority of users but they are considerations that must be taken into account when adopting virtualization technology. There are many virtualization software solutions available on the market; Oracle VirtualBox is one of these solutions.

Oracle VirtualBox

Oracle VirtualBox is an open source application available free of charge under the GNU General Public License (Oracle Corporation, 2013). Personal use covers the individual user, business owner, or organization. VirtualBox is a “dual licensed” product, meaning Oracle chooses the terms under which the code is licensed. Enterprise users are “encouraged” to purchase commercial licenses to receive access to enterprise features (Oracle Corporation, 2013).

Oracle VirtualBox offers cross-platform support; meaning it can be installed on most OSs. The current version runs on Windows (XP and later), Linux, Macintosh and Solaris host OSs on x86 AMD or Intel-based computers (Oracle Corporation, 2013). VirtualBox supports a number of guest OSs installed as VMs including, but not limited to, Windows (NT 4.0 and later), DOS/Windows 3.x, Linux (2.4 and 2.6), Solaris and OpenSolaris, OS/2, and OpenBSD (Oracle Corporation, 2013). Guest VMs can be 32-bit or 64-bit OSs.

Oracle recommends at least 512 MB of RAM to run but recommends more RAM for running guest OSs like Windows. The VirtualBox software itself requires relatively little hard disk space at 30 MB, but each VM requires a larger amount of hard disk space depending on the OS and its allocated use. The VM disk image can be set as fixed-size or dynamically allocated. Fixed-size images will grow only to the allocated size; for example, a 10 GB image will top out at the maximum capacity and alert the user that there is no more space. Dynamically allocated images are more flexible; starting out as a small file and growing to the allocated size. The benefit is that this type of VM image will take up very little space initially. The drawback is that it requires more computer resources to expand the file (Oracle Corporation, 2013). Once the virtual disk image has been created, the guest OS can be installed from an ISO image mounted from the virtual CD-ROM drive. The guest OS can be a licensed product or an open source OS, Linux Ubuntu for example. The combination of open source virtualization and open source OSs create a virtualization solution that is free of monetary cost, a factor that contributes to the increased use of virtualization.

After the OS is installed, the Guest Additions package can be installed. The Guest Additions feature allows for the sharing of drives, files and peripherals (Fitzpatrick, 2010). The Guest Additions are necessary for capturing the mouse; if this package is not installed the mouse

has to be captured and released every time the user moves from host to guest and vice versa. This package also allows the VM to be viewed as a seamless window on the host OS. The Guest Additions package facilitates the use of the guest VM (Oracle Corporation, 2013).

VirtualBox has a Snapshots feature that allows the user to save the state of a VM and revert to it at a later date even after changes have been made; any changes or additions made to the VM disappear after restoring to a previous snapshot (Oracle Corporation, 2013). A snapshot consists of components: a complete copy of the VM settings, including the hardware configuration, the complete state of all the virtual disks attached to the machine, and, if the snapshot was taken while the machine was running, the memory state of the machine (Oracle Corporation, 2013). The settings are stored in the machine configuration XML text file. The virtual disk itself is not restored; VirtualBox creates differencing images containing only the changes since the snapshot were taken. When the snapshot is restored, VirtualBox dumps the differencing image and goes back to the previous state (Oracle Corporation, 2013). In this way, all changes to the machine are undone when reverted. This is helpful for testing, as any mistakes can be undone by reverting back to an earlier stable state. The same is true when using a VM as a digital forensics lab; after every use the VM is reverted so that the next session is at a "clean" state. The snapshot feature is also beneficial as an anti-forensics technique for destroying or obfuscating evidence.

Implications of Virtual Machines and Cybercrime

Marcus K. Rogers (2006) defines anti-forensics as "attempts to negatively effect [sic] the existence, amount and/or quality of evidence from a crime scene, or make the analysis and examination of evidence difficult or impossible to conduct." There are four varieties of anti-forensic techniques: data hiding, artifact wiping, trail obfuscation and attacks against forensic

process or tools (Rogers, 2006). Of these, deleting a VM or reverting a VM back to a snapshot of the system before illicit activity was conducted fall into the varieties of artifact wiping and trail obfuscation. Virtual machines are simple to create and just as simple to delete, they are portable (VMs can be installed on and run from a USB drive), and the snapshot feature can be used as plausible deniability (it is easier for a user to revert to a snapshot than to attempt to delete all trace of a VM from the host). VMs offer a full computing environment where a user can browse the Internet, download and save files, create files, send and receive email and instant messages and any other tasks that can be performed in the installed OS, all packaged in an environment that can be physically removed from the machine if installed on a USB device or obliterated. This leaves little or no evidence behind of their activities for a digital forensic analyst to discover (Garfinkel, 2007).

The increased use of VMs has opened a new avenue of research in the field of digital forensics. VMs are used as forensic tools for analysis; there are many virtual platforms pre-built with forensic tools. Techniques have been developed to acquire VM images for analysis, but very little research has been dedicated to the problem of recovering evidence of activity after a VM has been deleted or reverted. The common notion is that due to the nature and structure of a VM, once the environment has been deleted or reverted, any trace of activity is unrecoverable due to the isolation of the guest and host systems (Swanson & Williams, 2008). This is a dilemma for digital forensic analysts and a benefit for cyber criminals. Through research, the goal is to find a solution that advances the field of digital forensics.

Literature Review

The field of digital forensics has responded to the proliferation of VMs by identifying the benefits and drawbacks that affect forensic investigations. Cosimo Anglano is an associate

professor of the Computer Science Department of the Universita' del Piemonte Orientale "A. Avogadro". He possesses a Ph.D. in Computer Science and is the founder and director of the "Centro Studi sulla Criminalita' Informatica" (Center for Studies on Cybercrime). He has published more than 60 papers in international journals. His current research focuses on Cloud Computing and digital forensics (Anglano, Cosimo anglano home page, 2012). Anglano (2010) discusses these issues and the concerns for digital forensic investigations:

> The proliferation of these technologies will result, in the near future, in an increasing number of illegal or inappropriate activities carried out by means of virtual machines, or testing virtual machines, rather than physical ones. (p. 424)

Anglano proceeds to identify the need for more comprehensive research into and development of tools and methodologies tested to the unique problems presented by VMs (2010). These concerns have been addressed by experts in the field of digital forensics; several virtual machine platforms based on Linux have been developed as forensic examination platforms which are pre-configured with common forensic tools (Anglano, Forensic implications of virtualization technologies., 2010). The use of virtualization to run an acquired forensic image as a VM for analysis is another way that virtualization has been adopted as a forensic solution. Tools and techniques have been developed that can be used to analyze an acquired VM image. In spite of these developments, Anglano recognizes the lack of literature that presents a comprehensive discussion about the challenges that VMs present, including the use of VMs as anti-forensic tools (Anglano, Forensic implications of virtualization technologies., 2010).

Doctor Iain Swanson & Doctor Patricia Williams, Professors of Computer and Information Science at Edith Cowan University of Australia also address the likelihood of the use of VMs as anti-forensic tools. Anti-forensics seeks to destroy or hide data in order to decrease the chances

of a forensic analyst finding evidence of criminal activity through data hiding, artifact wiping, obfuscation, and direct tool attacks (Swanson & Williams, 2008). Using a VM in this capacity, any activity that a user engages in that would normally leave behind digital evidence on the hard drive would be confined to the virtual hard drive image. Due to the isolation of the VM from the host, the assumption is that no evidence of activity would be found on the host hard drive, except for files indicating that a virtualization software solution may be or may have been installed (Swanson & Williams, 2008). Deleting or altering that image would cause the evidence to become unrecoverable or insufficient for analysis (Swanson & Williams, 2008).

The deletion of a VM is commonly understood to be an effective way to eliminate evidence (Shavers, 2008). Brett Shavers, author of X-Ways Forensics Practitioners Guide, describes a typical scenario in which a deleted file usually goes to the recycle bin. In this scenario, the files can often be recovered for analysis. Since VMs are large files, file size deletion limits may prevent the file from going to the recycle bin and may instead be deleted directly by the system. Files deleted in this way may be recoverable, but the fragmentation of the VM will most likely make full recovery and analysis impossible (Shavers, 2008). The experiments detailed in the methodology and testing sections of this research, as well as the analysis of evidence acquired after the testing phase will determine the veracity of this notion.

Methodology

In an effort to ascertain the validity of these assumptions, an experiment was designed that would test a computer on which Oracle VirtualBox was installed. VirtualBox offers two options for discarding a VM; the remove option which only removes the VM from the manager, or to delete the VM which deletes all the files associated with the VM. The process of reverting a VM to a snapshot is not a method for discarding a VM but for discarding activity on a VM. The

testing process required the creation of three Linux Ubuntu VMs: one that would be removed from the VirtualBox application, one that would be reverted to a snapshot, and one that would be deleted. Actions were taken on each VM which consisted of file downloads, file creation and web browser activity.

While there was no defined scenario, the analyst defined an assumption to work from during the testing phase and analysis phase as to the knowledge level of the hypothetical user. This was required to limit the scope of the research. The assumption was that the hypothetical user in this case was more familiar with computer technology than the average user; familiar enough to know about virtualization technology, its uses and benefits for obfuscating evidence, and how to install and operate open-source OSs like Linux. Although this user possessed above-average knowledge of the technology, their knowledge of how to dispose of evidence was limited, with no in-depth knowledge or ability to make significant changes to the fundamental structure of the host OS in order to dispose of evidence. The obfuscation was limited to disposing of the evidence present in the VM environments by regularly reverting to a saved state after use, or by deleting them with the assumption that any evidence of their activity would be forensically unrecoverable.

Tools

Several tools were used to monitor the state of the test computer at each phase of testing and document changes in the registry and file system. The documentation that these tools provided was used as a starting point in determining what types of files are created by VirtualBox, what files were changed or deleted during the testing phase, and the locations of potential evidence left behind. All tools used during testing were launched from an external Universal Serial Bus (USB) flash drive; this minimized the impact that the testing process had on

the test machine as only the required VirtualBox and OSs needed to run the scenario were installed directly on the machine. After the testing phase was accomplished, images of both the RAM and the hard drive image were acquired for analysis.

Regshot. Regshot is an open-source application that compares two snapshots taken of the Windows registry before and after system changes or software installation (Google Code, 2013). Regshot is supported on Windows 2000, XP, Vista, and 7 (both 32-bit and 64-bit) OSs. It is a valuable tool for malware research and testing. Regshot version 1.8.3-beta2 was downloaded onto a 16 GB USB drive by the analyst before testing (Google Code, 2013). Regshot snapshots were generated before and after each phase of testing and compared using the tool's Graphical User Interface (GUI). The comparison results were saved as a series of text files that would be reviewed during analysis.

FolderChangesView. FolderChangesView is a simple tool developed by NirSoft that monitors a designated folder or disk drive and lists every filename that is being modified, created, or deleted while the folder is being monitored (NirSoft, 2013). The application records the changes and the function can be toggled using the ubiquitous play and stop icons on the GUI. FolderChangesView version 1.50 was downloaded onto the 16 GB USB drive by the analyst before testing and used primarily to document files and directories that were created during testing (NirSoft, 2013). The application was set to record changes during each phase of testing and the results were saved to text files for later analysis.

WhatChanged. WhatChanged is a system utility developed by VTask Studios that scans for modified files and Windows registry entries (VTask Studio, 2013). A baseline snapshot is generated and later compared to a second snapshot taken after changes are made to the system. WhatChanged v1.07 was downloaded onto the 16 GB USB drive by the analyst before testing

(VTask Studio, 2013). This was primarily used as a backup documentation solution in case one of the other two applications failed.

7-Zip portable. 7-Zip is a file archiver (compression) utility for Windows. The portable version is offered by PortableApps.com and is optimized to run from a USB flash drive. PortableApps is a platform that offers portable versions of popular applications that can be installed directly to a USB instead of being installed on a computer. This allows flexibility and ease-of-use, as a user can have their most-used applications with them and run them on any computer (PortableApps.com, 2013). 7-Zip was downloaded onto the 16 GB USB drive by the analyst before testing (PortableApps.com, 2013). 7-Zip was used to extract 7z archive files during the course of the testing process.

Testing Results

The first phase of testing was to prepare the test environment. This phase included a clean install of the host OS, with all drivers and updates installed. The test system was a Toshiba Satellite A215-S4757 laptop computer running Windows Vista Business Service Pack 2. The processor was an AMD Turion™ 64 X2 Mobile Technology TL-56 @ 1.80 GHz. The system had 2 GB RAM. The system time zone was Eastern Standard Time and was validated as the correct time and date. The analyst installed the tools which were previously downloaded onto the 16 GB USB flash drive attached to the analysis computer and launched the Regshot, FolderChangesView and WhatChanged applications on the test computer.

The second phase of testing involved the download and installation of the VirtualBox application as well as the Linux Ubuntu 12.10 and Ubuntu 13.04 OS files. Before the download and installation of the Oracle VirtualBox application, the analyst started the recording feature of the FolderChangesView application, and generated a snapshot of the system state using the

Regshot application. Once the file and Windows registry monitoring was underway, the analyst downloaded Oracle VirtualBox v4.2.18 in the test system's Internet Explorer web browser and installed on the test system (Oracle Corporation, 2013). The analyst generated another Regshot snapshot after VirtualBox was installed and generated the first FolderChangesView log which was saved as a text file on the USB flash drive.

The analyst downloaded the Ubuntu 12.10 VDI image in the test system's Internet Explorer web browser and saved in the Downloads folder of the test system (VirtualBoxes-Free VirtualBox ® Images, 2013). Virtual disk images (VDI) files are proprietary VirtualBox files that store all virtualized data and virtualized OS settings in one file. The VDI file is the virtualized physical disk that VirtualBox uses for each VM (File-Extensions.org, 2013). VDI files are mounted in VirtualBox, emulating a physical drive. The Ubuntu 12.10 VDI downloaded was preconfigured, therefore allowing the build of a new VM without having to go through the steps required to set up a VM from an OS image file, as opposed to building the VM using an Ubuntu 12.10 ISO file. The analyst extracted the file to the desktop using the 7zip Portable application launched from the 16 GB USB flash drive.

The analyst generated a Regshot snapshot before downloading the image file needed to build the Ubuntu 13.04 VM titled ubuntu-13.04-desktop-i386.iso. The analyst downloaded the image file in the test system's Internet Explorer web browser and saved it in the Downloads folder of the test machine (Ubuntu, 2013). After the download, the analyst generated another Regshot snapshot, sent the comparison output to a text file and saved to the 16 GB USB flash drive for later analysis.

The third phase of testing entailed the creation of the VMs. The analyst generated a Regshot snapshot before beginning the installation of the VMs and launched the VirtualBox

application on the test machine. Figure 1 shows the Oracle VM VirtualBox Manager before the creation of any VMs. The pane on the left below the toolbar is empty; this area is where the VMs will be listed and launched. The center area of the VirtualBox Manager will show the details of the selected VM once the manager is populated. The blue "New" button is the only option available, and it will launch the VM creation wizard.

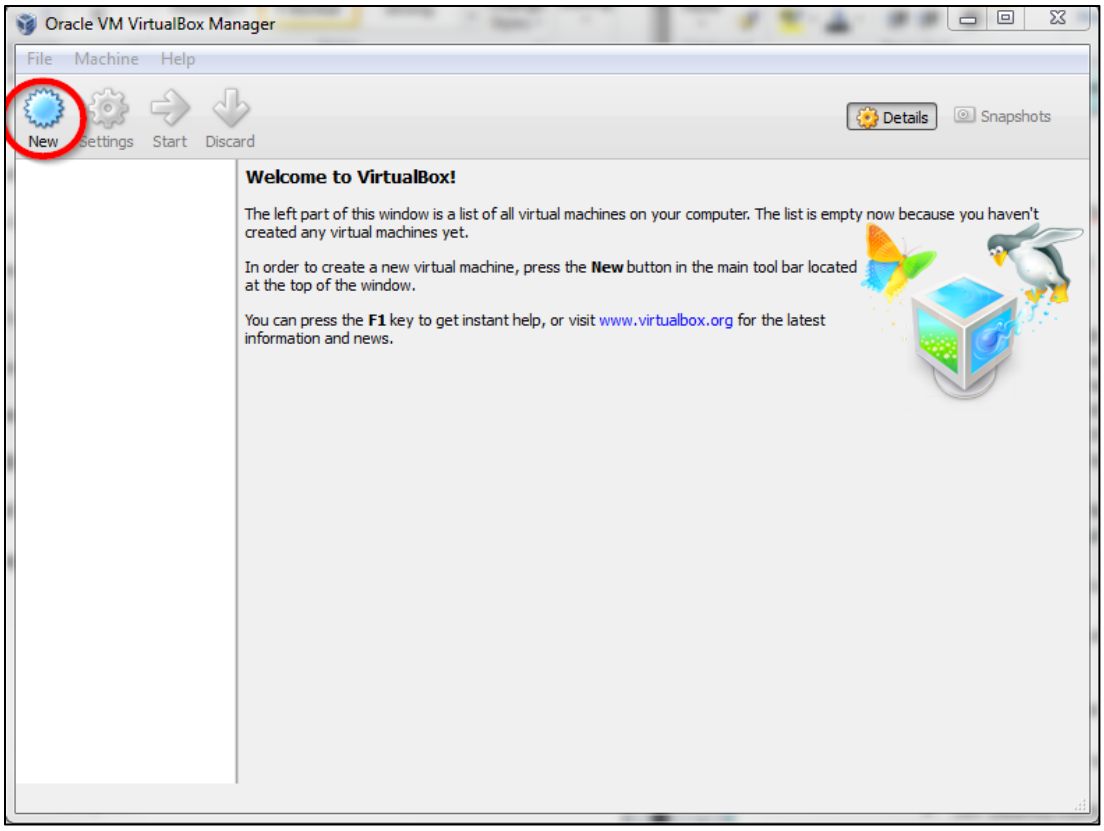


Figure 1. Oracle VM VirtualBox Manager.

In the VirtualBox Manager menu, the analyst selected the blue "New" button. The analyst created the VMs using the default or recommended settings provided in the wizard. The analyst created a VM for the Ubuntu 13.04 guest OS, maintaining the memory size was at the default 512 MB setting and set the virtual hard drive storage to the recommended 8 GB size. The analyst created the virtual hard drive as a fixed sized VDI file and stored it in the

"C:\Users\Test\VirtualBox VMs\Ubuntu 13.04" directory. This step created the virtual hard drive, but the actual OS was yet to be installed at this phase.

Under the Settings button, the analyst attached the Ubuntu 13.04 ISO image as a virtual CD/DVD drive. The VirtualBox application allows image files to be mounted as physical media. The Ubuntu 13.04 ISO file was treated as if it was on a CD or DVD installation disc and was installed from that media. Figure 2 shows the steps that were taken to install the OS that would be used on the new VM.

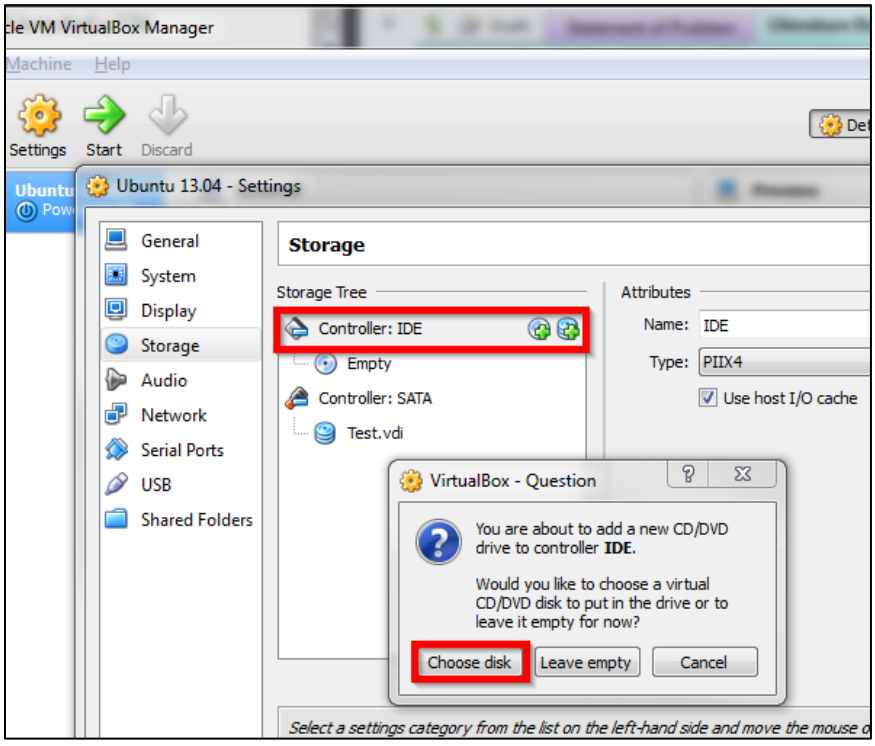


Figure 2. Loading the ISO image.

Once the ISO image was attached, the analyst launched the Ubuntu 13.04 VM. The OS booted successfully from the virtual CD/DVD drive. Once the boot process was completed, the Ubuntu 13.04 desktop was displayed and offered two options; to try Ubuntu or to install it. The analyst initiated the installation process. This process wiped the virtual hard drive and installed the Ubuntu OS. Once the installation process was complete, the analyst restarted the VM and

logged into the Ubuntu 13.04 OS to validate that it was created correctly. The analyst generated a Regshot snapshot after the creation of the Ubuntu 13.04 VM, sent the comparison output to a text file which was saved it to the 16 GB USB flash drive for later analysis.

The creation of the Ubuntu 12.10 VM was much simpler due to the fact that the VDI file, the virtual hard disk, was already created and formatted. Instead of selecting the "Create a virtual hard drive now" option as was the case with the creation of the Ubuntu 13.04 VM, the analyst selected the "Use an existing virtual hard drive file" option in the wizard. The analyst used the drop-down menu below the option to navigate to the VDI file previously saved on the desktop. Figure 3 shows the option used to create a VM from the Ubuntu 12.10 VDI file.

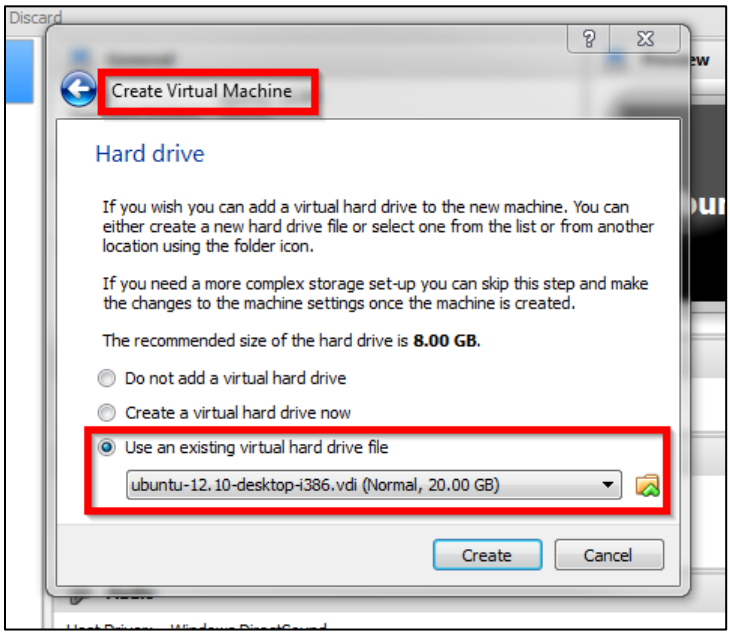


Figure 3. Creating a Virtual Machine hard drive from a VDI image file.

Once the Ubuntu 12.10 VM was created, the analyst launched the VM and logged into the Ubuntu 12.10 OS to validate that it was created correctly. The analyst generated a Regshot snapshot after creating the Ubuntu 12.10 VM, sent the comparison output to a text file and saved it to the 16 GB USB flash drive for later analysis. The analyst stopped the FolderChangesView application file recording, and the output was sent to a text file which was saved to the 16 GB

USB flash drive for later analysis. With the VMs created, the last phase of testing began. This phase of testing was comprised of generating user activity including web browser activity, file downloads from the Internet, and files created using a word processing application in the Ubuntu 12.10 and Ubuntu 13.04 guest OSs.

Test #1: Removal of a VM

This test began with the use of the Ubuntu 12.10 VM to test the removal of a VM. The analyst generated a Regshot snapshot, a WhatChanged baseline snapshot, and started the FolderChangesView file recording before testing began. The analyst launched the VirtualBox application and opened the Ubuntu 12.10 VM. For the test web history generation test, the analyst launched the Firefox web browser application from the desktop and entered "cyber crime laws florida" in the Google search bar. The analyst selected the first result and navigated to www.secureflorida.org/legalissues/computer_laws/. In a new tab, the analyst typed http://www.hacker10.com/ into the address bar and navigated to the website. For the downloaded images test, the analyst opened a new tab and entered "girl halloween costumes" into the Google search bar. In the results page, the analyst navigated to the Google Images tab and selected a random image, saving the image titled “61aUpZ7V1qL._SL1499_.jpg” in the Downloads folder. The analyst returned to the Google image results, selected another random image and saved the image titled “9195-main.jpg” in the Downloads folder. For the downloaded documents test, the analyst entered "girl halloween costumes .pdf" in the Google search bar and selected http://www.epilogsys.com/ScoutingWeb/Documents/Silver%20Proj%20Ideas.pdf from the results. The analyst downloaded a file titled "Silver Project Ideas.pdf" and saved it to the Downloads folder. The analyst saved the tabs as Favorites and closed Firefox.

For the document creation test, the analyst launched LibreOffice Writer and typed "I love Halloween!!" into the document, saved the document as “"halloween.odt" in the Documents directory and closed the application. Once the testing was completed, the analyst shut down the Ubuntu 12.10 VM. The analyst generated a Regshot snapshot, stopped the FolderChangesView application file recording and sent the comparison output of both applications to text files saved to the 16 GB USB flash drive for later analysis.

The analyst right-clicked on the Ubuntu 12.10 VM in the VirtualBox Manager and selected the “Remove” option. The Remove option launched a dialog box which offers two choices; to delete all files or to remove only. The analyst selected the “Remove Only” option from the dialog box. The Remove Only option does not delete the VDI file; it only removes the VM from the VirtualBox Manager. The VDI file and the VBOX definitions file remain in the VirtualBox Manager and can be added back into the VirtualBox Manager at any time. Figure 4 shows the remove option in the menu and Figure 5 shows the options for removing a VM.

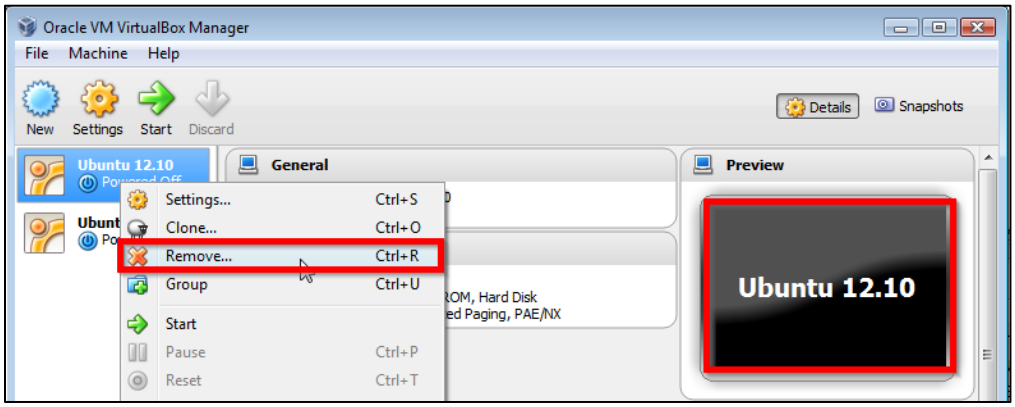


Figure 4. Removing Ubuntu 12.10.

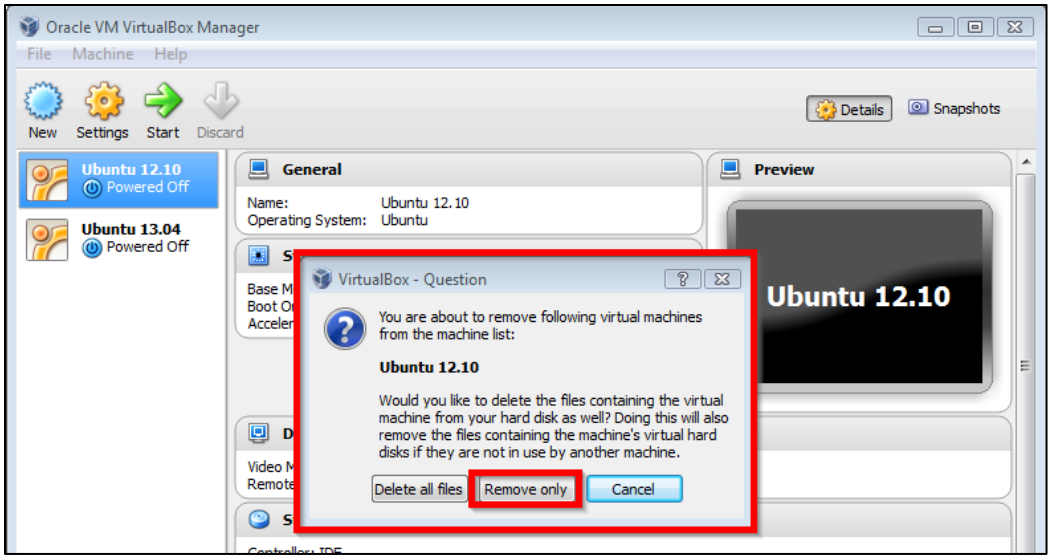


Figure 5. Removing the VM only option.

Test #2: Revert a VM to a Previously Saved State (Snapshot)

The next test performed was to save the running state of a VM (create a snapshot), create web browser history and files, and revert the VM to the snapshot. After the removal of the Ubuntu 12.10 VM from the VirtualBox Manager, the analyst created a new VM using the same VDI file and the same process as was used to create the Ubuntu 12.10 VM. The analyst titled this VM Ubuntu 12.10 2. After the creation of the VM, the analyst logged into the Ubuntu 12.10 2 OS and confirmed that the web browser history, downloaded files and favorites that were generated before removal were still present in their respective directories, validating that the removal of the VM did not change the VDI file. The analyst created a snapshot of the running state of the VM by selecting the "Machine" tab at the top of the VirtualBox window and selecting "Take Snapshot". Snapshots can be saved with the VM either running or powered off; since this snapshot was taken in the running state, the restored VM will resume at exactly the point when the snapshot was taken (Oracle Corporation, 2013). The snapshot was saved as Snapshot 1. Figure 6 and Figure 7 show how the snapshot of the VM running state was created and saved.

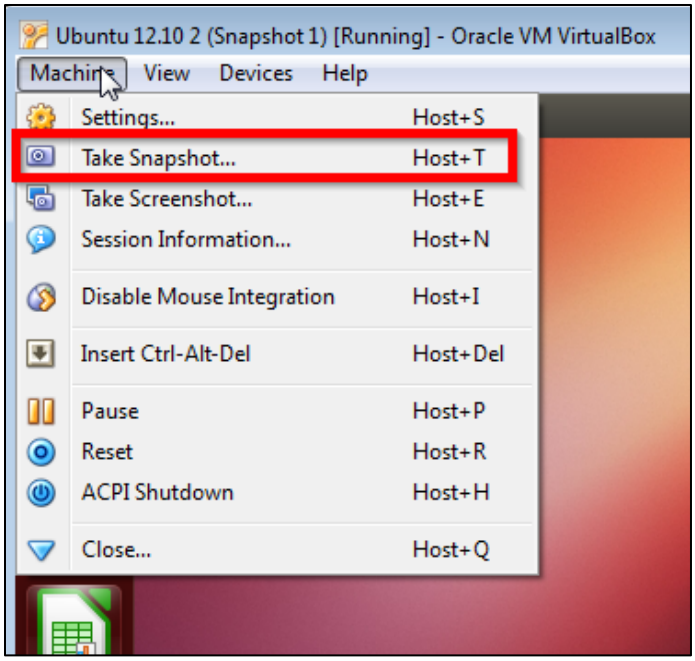


Figure 6. Take a snapshot.

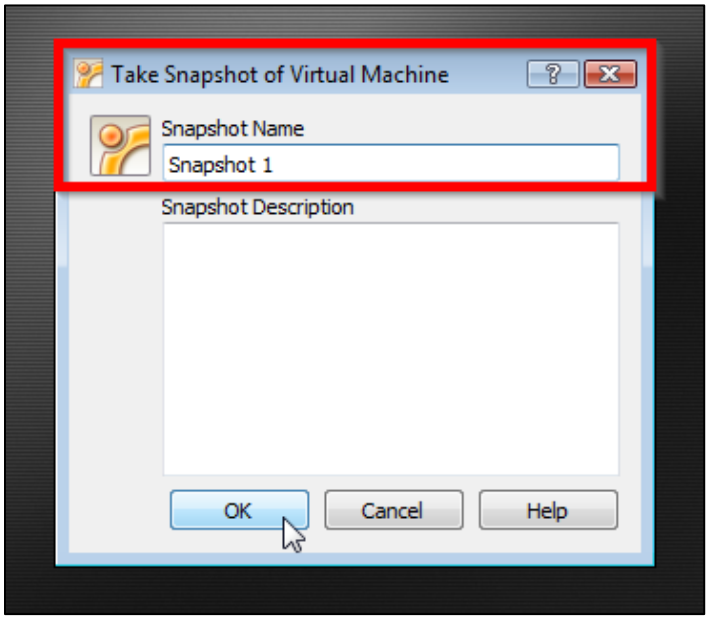


Figure 7. Save a snapshot.

Once the state of the machine was saved as a snapshot, the analyst launched the Firefox web browser application. To test the web history and downloaded images test, the analyst entered the search term "girl halloween costumes", navigated to the Google Images tab in the results page and selected a random image. The analyst saved the image as "girl-vampire-

costume.jpg" in Downloads folder. For the downloaded documents test, the analyst entered the search term "halloween costumes .pdf" in the Google search bar and navigated to http://lmc.gatech.edu/~cpearce3/PearcePubs/LudicaDress-Up.pdf and saved the file "LudicaDress-Up.pdf" to Downloads folder. The analyst closed the Firefox application, opened the LibreOffice Writer application, an open-source word processing application that comes preinstalled on Ubuntu OSs, and typed "I love candy" in the document. The document was saved as "halloween 1.odt" on the desktop (.odt is the file type native to LibreOffice Writer). The analyst closed the LibreOffice Writer application and shut down the Ubuntu 12.10 2 VM.

From the VirtualBox Manager, the analyst reverted the Ubuntu 12.10 2 VM by selecting the “restore snapshot” icon. The dialog box that opened warned the analyst that the current state will be permanently lost. Figure 8 shows the location of the Restore Snapshot icon. Selecting this icon provides options for restoring a saved snapshot.

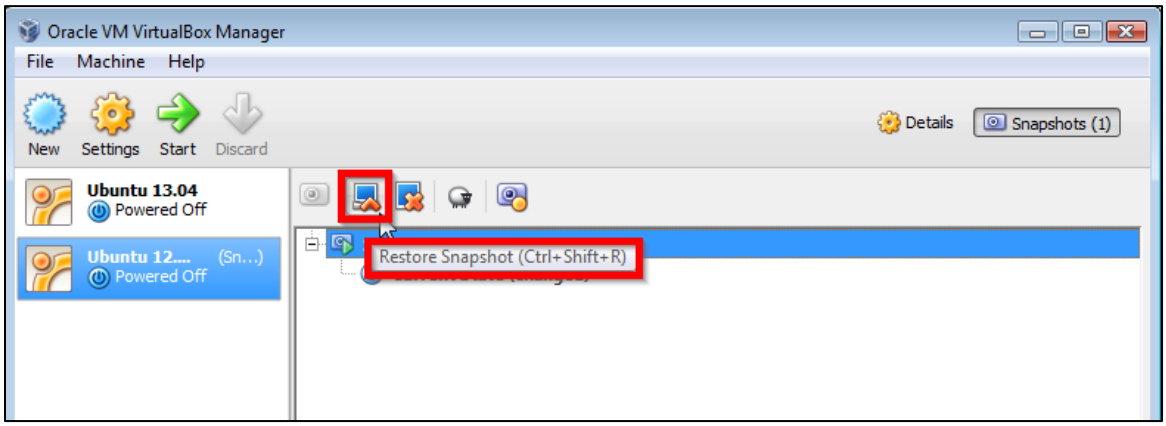


Figure 8. Reverting to snapshot.

Figure 9 shows the dialog box that is displayed when the “Revert snapshot” icon is selected. Note the warning that the current state of the VM will be permanently lost if no backup is created for the current state.

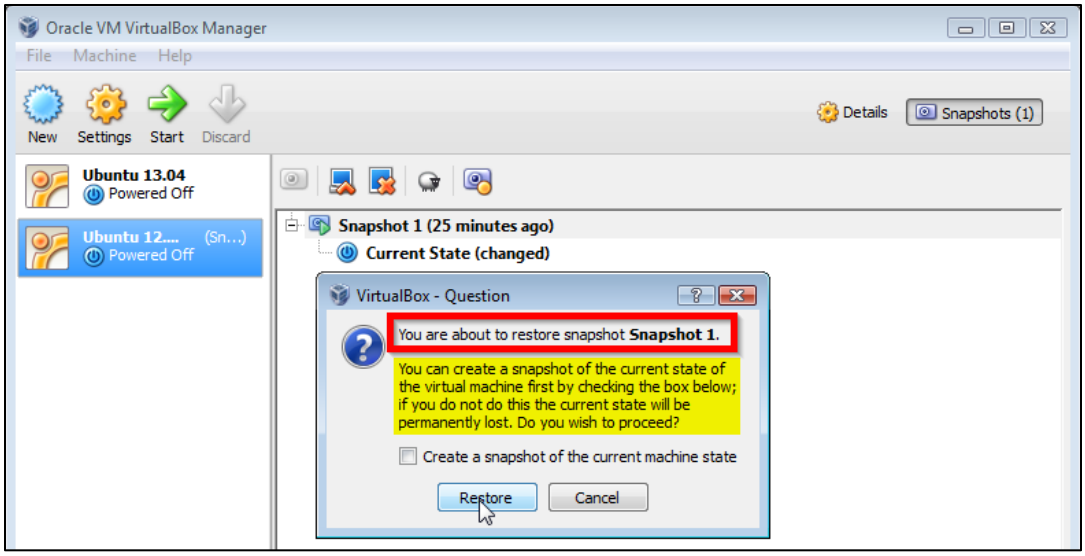


Figure 9. Restoring Snapshot 1.

The analyst reverted the VM back to Snapshot 1, launched the Ubuntu 12.10 2 VM and searched for the files that were created prior to the restore. The analyst confirmed that the "girl-vampire-costume.jpg", "LudicaDress-Up.pdf" and the "halloween 1.odt" files were not located in the directories in which they were originally saved to. The analyst also confirmed that the web history from the prior activity, which included the "halloween costumes .pdf" Google search and the http://lmc.gatech.edu/~cpearce3/PearcePubs/LudicaDress-UP.pdf search result were not saved. Only the files and history created in the Ubuntu 12.10 VM when Snapshot 1 was saved remained.

Test #3: Delete a Virtual Machine

The last test was performed on the Ubuntu 13.04 VM. The analyst launched the VirtualBox application and opened the Ubuntu 13.04 VM. The same testing process was performed as was performed on the previous VMs. For the web history test, the analyst launched the Firefox web browser application from the Ubuntu 13.04 desktop and entered “cult of the dead cow” in the Google search bar. The analyst selected the first search result and navigated to w3.cultdeadcow.com. In a new tab, the analyst typed www.petrifiedtruth.com into the address

bar and navigated to the website. For the downloaded images test, the analyst opened a new tab and entered “costa rica” into the Google search bar. In the results page, the analyst navigated to the Google Images tab, selected a random image and saved it as “beach.jpg” in the Pictures directory. The analyst returned to the “costa rica” search results, selected the link of a map of Costa Rica and navigated to http://www1.internationalliving.com/sem/country/costa-rica/google/search/kw-lp.html?gclid=CLqBmoPBtboCFUdk7AodI3MAng. For the downloaded documents test, the analyst entered “linux forensics” in the Google search bar and selected http://www.blackhat.com/presentations/bh-usa-03/bh-us-03-willis-c/bh-us-03-willis.pdf from the results. The analyst downloaded the PDF file titled “Forensics with Linux 101 or How to do Forensics For Free” and saved the file as bh-us-03-willis.pdf in the Downloads folder. The analyst saved all the tabs as “Faves” and closed Firefox.

For the document creation test, the analyst launched LibreOffice Writer, typed "This is a test Password: junglegym" into the document, saved the document as “Gym.odt” in the Documents directory and closed the application. Once the testing was completed, the analyst shut down the Ubuntu 13.04 VM. The analyst generated a Regshot snapshot, stopped the FolderChangesView application file recording and sent the comparison outputs to text files which were saved to the 16 GB USB flash drive for later analysis.

After completing the file creation test, the analyst generated a Regshot snapshot and deleted the Ubuntu 13.04 VM. The analyst right-clicked on the Ubuntu 13.04 VM in the VirtualBox Manager and chose Remove from the menu. Figure 10 shows the “Remove” option in the VM menu.

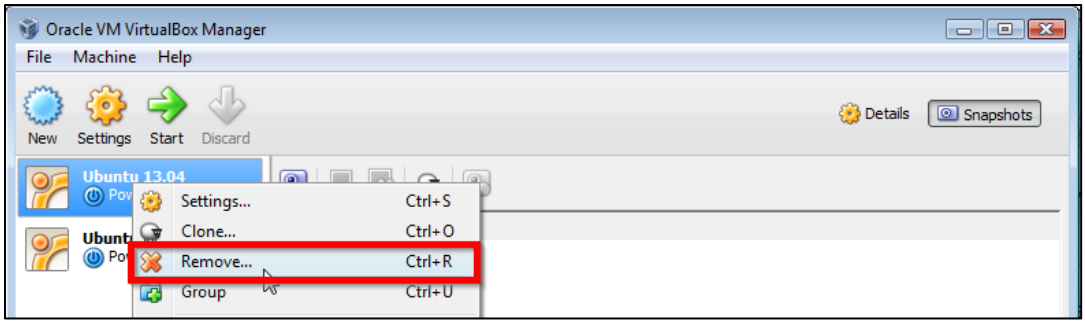


Figure 10. Removing the Ubuntu 13.04 VM.

For this test, the "Delete all files" option was selected. After the VM was deleted, another Regshot snapshot was generated and the comparison output was sent to a text file which was saved to the 16 GB USB flash drive for later analysis. Figure 11 shows the dialog box that is displayed when the "Remove" option is selected from the menu. Note the warning that the "Delete all files" option will remove all the VM files.

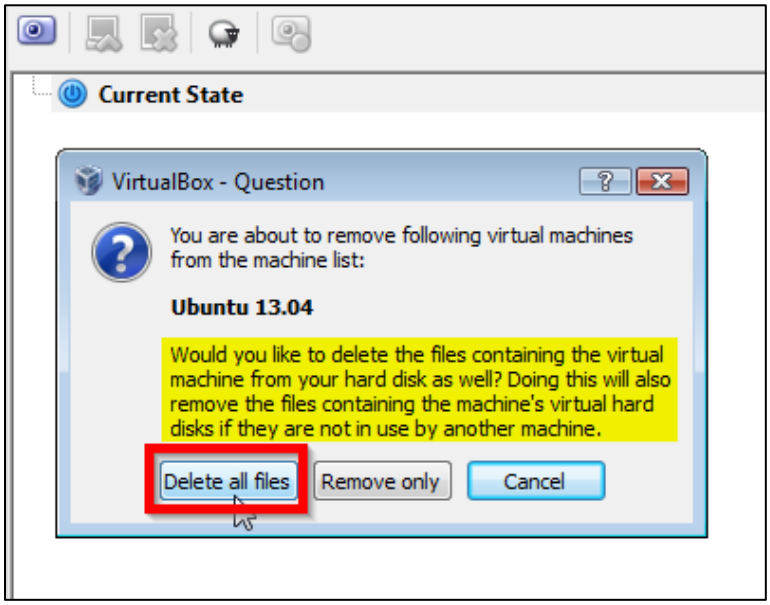


Figure 11. Deleting all files.

After the deletion of the Ubuntu 13.04 VM, the analyst confirmed that the VDI and VBOX files were no longer present in the VirtualBox VMs/Ubuntu 13.04 directory. The analyst closed the VirtualBox application, generated a Regshot snapshot and saved the comparison output file to a text file for analysis. The analyst generated a comparison file in WhatChanged and saved it

to a text file for analysis. The analyst stopped the FolderChangesView application and saved the results to a text file for analysis. This concluded the testing phase. In the next phase of the research, the analyst collected and examined the data from the testing phase in an effort to recover evidence of the user-generated activity conducted on the three VMs.

Data Collection

The data collection process began with the acquisition of the volatile data, RAM and hard drive of the target computer. Sound forensic practices were applied during data preservation, collection, recovery and analysis. The acquisition was performed on the test computer with the machine still powered on from the testing phase. The analyst used Helix 3 Pro to acquire the volatile information including network, processes, drivers, environmental variables, installed applications and user information. Helix 3 Pro is a forensic tool that is launched from a CD. It contains both a live and a bootable environment; the live environment is a GUI that contains a set of tools for forensics and the bootable side contains a Linux environment that is built on the Ubuntu platform (E-fense, 2013). The analyst inserted the Helix 3 Pro CD into the CD drive and launched the live environment. The volatile data output that the analyst acquired was sent to a file titled VolatileData.txt and saved on the 16 GB USB drive.

The analyst used Forensic Toolkit® (FTK) Imager Lite to acquire images of the physical RAM and the hard disk from the target machine. FTK Imager Lite is a forensic acquisition tool created by AccessData that can be run from a CD or USB flash drive (AccessData, 2013). The analyst saved the RAM as “Memdump.mem” on a separate 16 GB USB flash drive than the one that was loaded with the analyst’s tools. The analyst acquired the disk image in the EnCase E01 format and saved image file as “satellite_disk_image.E01” on a 250 GB external hard drive.

The analyst used WinMD5 Free to generate and compare the hash values of the "VolatileData.txt", "Memdump.mem" and "satellite_disk_image.E01" evidence files for integrity after analysis. The hash values of the files were generated in order to validate that the files had not been altered after the analysis process. WinMD5 Free is a no-cost utility that can compute MD5 hash values for files. It is compatible with Microsoft Windows 98 through Windows 7 (WinMD5, 2009). A preservation copy of each file was created from the original evidence files and saved on a 500 GB external hard drive. A working copy of each file was created from the preservation files and saved onto the examiner's machine for analysis.

The Regshot, FolderChangesView and WhatChanged log files that had been generated during the testing process captured registry and file changes during each phase of the testing process. The log files had been saved as text files on the 16 GB USB drive. The analyst created working copies of the text files and the data was copied to Windows Excel spreadsheets for analysis.

Analysis

Change Management Logs

The analyst began this phase with the review of the change management log files in order to identify which files and directories were created by the Oracle VirtualBox application. The Regshot log files were reviewed to find and identify files and registry keys that were created or changed when VirtualBox was downloaded and installed. Evidence of the installation of VirtualBox was referred to as being located in several registry keys, including the "HKLM\SOFTWARE\Classes", "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion", "HKLM\SOFTWARE" and "HKLM\SYSTEM\ControlSet001\Control\DeviceClasses" keys. VirtualBox supports various disk image formats, including .hdd, .ova, .ovf, .vbox, .vdi, .vhd and

.vmdk. These file settings were listed in the Regshot log entries as being located in the "HKLM\SOFTWARE\Classes" registry key. The VirtualBox application was documented in the Regshot log entries as having been installed as was documented in the "C:\Program Files\Oracle\VirtualBox" as well as the "C:\Windows\system32\VBoxNetFltNobj.dll" directories. The Regshot log entry"HKU\S-1-5-21-1551165937-3217561686-2581226915-1000\Software\Oracle\VirtualBox\Install" documented the value "installed" in the registry key. The "C:\Users\SABINT~1\AppData\Local\Temp\VirtualBox\" directory Regshot log entry also documented that the VirtualBox application was installed. The Regshot log entries documented the version of VirtualBox that was installed; Oracle VM VirtualBox 4.2.18, in the "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall" registry key. The install date of the VirtualBox application was shown in the Regshot log files as being located in the "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall" registry key and was validated against the date that was recorded during testing.

The FolderChangesView log files documented evidence of VirtualBox in the Temporary Internet Files directory. The Windows registry key "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\A56400C238AEB9D449281908BFBE4DCA\InstallProperties\URLInfoAbout:http://www.virtualbox.org" was documented in the FolderChangesView log files as the location from which the VirtualBox application was downloaded. The location to which the VirtualBox application was installed was documented in the log files as "HKLM\SYSTEM\ControlSet001\Control\Session Manager\Environment\VBOX_INSTALL_PATH: "C:\Program Files\Oracle\VirtualBox\". The "VirtualBox-4.2.18-88781-Win.exe" file was documented in the log files as being located in the "C:\Users\Test\Downloads" directory, showing that the application was downloaded. Evidence

was documented in the log files in the “C:\Users\Test\AppData\Local\Temp” directory. The “C:\Windows\Prefetch\VIRTUALBOX-4.2.18-88781-WIN.E-49C7DFE9.pf” location was documented that the application was run on the test machine. The VirtualBox drivers were documented in the logs as having been installed in the “C:\Program Files\Oracle\VirtualBox\drivers” directory and in the “C:\Windows\System32\DRVSTORE\VBoxUSBMon_85ED5D36B5EA99EC0D0D9654290FE965158CB4AB” directory. The VirtualBox installation process was also documented in the logs; the “C:\ProgramData\Microsoft\Windows\Start Menu\Programs” directory showed where the VirtualBox application had been installed. Following the installation of the VirtualBox application on the test machine, Ubuntu 13.04 and Ubuntu 12.10 were added as VMs. The Regshot, FolderChangesView and WhatChanged applications document that phase of testing.

Evidence that Ubuntu 13.04 was downloaded from the www.ubuntu.com website was documented in the “C:\Users\Test\AppData\Local\Microsoft\Windows\Temporary Internet Files\Low\Content.IE5” directory. Evidence of the ubuntu-13.04-desktop-i386.iso file that was downloaded was documented in the “C:\Users\Test\Downloads” directory. The process of building a VM from the ISO file created the directory “C:\Users\Test\VirtualBox VMs\Ubuntu 13.04”. In this directory, the VM-building process created the “Ubuntu 13.04.vdi” file. The VDI file was the guest hard drive container or the virtual hard drive that VirtualBox uses as storage (Oracle Corporation, 2013). The VM-building process also created the “Ubuntu 13.04.vbox” file in the “C:\Users\Test\VirtualBox VMs\Ubuntu 13.04” directory. The VBOX file is the XML settings file for the virtual machine, this file stores all the virtual machine files (Oracle Corporation, 2013). These directory locations are the default locations for the VirtualBox files, but it is important to note that the default virtual machine folder can be changed by the user

(Oracle Corporation, 2013). Another file that was created in the default directory is the "Ubuntu 13.04.vbox-prev" file. This file is the backup file for the associated VM; a corrupted VBOX file can be replaced with this one (after removing the –prev portion of the file name) if no snapshot was saved (Moussat, 2012). Configuration log files labeled "VBox.log" were created in the "C:\Users\Test\VirtualBox VMs\Ubuntu 13.04\Logs" directory. These are created when VirtualBox starts up a VM and contain information about the VM configuration and runtime events; every time the VM is run, the last configuration file will be renamed "VBox.log.1", up to .3 (File-Extensions.org, 2013) (Oracle Corporation, 2013). Entries in WhatChanged documented that the Ubuntu 13.04 VM was run twice; the "VirtualBox VMs\Ubuntu 13.04\Logs" directory documented "VBox.log" and "VBox.log.1" files.

The Ubuntu 12.10 VM was not created from an ISO file as was Ubuntu 13.04; instead it was created by opening a pre-configured VDI image file. In this scenario, the configuration steps taken to create the "Ubuntu 13.04.vdi" file had already been completed, all that was needed was for the VDI file to be added to a new VirtualBox VM. This automatically added the VM into the VirtualBox Manager. The entry "C:\Users\Test\AppData\Local\Microsoft\Windows\Temporary Internet Files\Low\Content.IE5\Y0CS4H8L\homepage-12.10-launch-ubuntu[1].png,14849" in the WhatChanged logs documented that Ubuntu 121.10 was downloaded from the Internet. Evidence that the file "ubuntu-12.10-desktop-i386.7z" was saved to the desktop was documented in "C:\Users\Test\Desktop\ubuntu-12.10-desktop-i386.7z". The entry "C:\Users\Test\Desktop\ubuntu-12.10-desktop-i386" documented that the .7z file was unzipped to the Desktop. Entries in the FolderChangesView logs show that the unzipped file contained the necessary files needed to create the VM; "C:\Users\Test\Desktop\ubuntu-12.10-desktop-i386\ubuntu-12.10-desktop-i386.vbox" and "C:\Users\Test\Desktop\ubuntu-12.10-desktop-

i386\ubuntu-12.10-desktop-i386.vdi". The entry "C:\Users\Test\VirtualBox VMs\Ubuntu 12.10\Logs\VBox.log,80805" documented that the Ubuntu 12.10 VM was booted, generating a configuration log. The Ubuntu 12.10 VM was run 3 times, as is evidenced by entries showing "VBox.log", "VBox.log.1" and "VBox.log.2".

The next testing phase after the Ubuntu 13.04 and Ubuntu 12.10 VMs were created was the generation of user activity within the VMs. The Ubuntu 13.04 and Ubuntu 12.10 VMs had user-generated activity performed on each. The browser history, downloaded files, and application files that were created during this phase do not appear anywhere in the change management logs. This initial analysis was important, as it facilitated an expansion of the analysis strategy in the forensic examination of the RAM and hard drive image.

As was documented during testing, each VM was chosen for a specific obfuscation test. The Ubuntu 12.10 VM was removed from the VirtualBox Manager but the VDI and VBOX files remained in the default directory. The "Ubuntu 12.10.vdi" file was used to create a new VM, labeled Ubuntu 12.10 2. The creation of this VM was documented in the change management logs. The "C:\Users\Test\VirtualBox VMs\Ubuntu 12.10 2" log entry documented that the folder for the VM files was created. The files generated during the creation of the VM were listed in entries in the FolderChangesView and WhatChanged logs; "Ubuntu 12.10 2.vbox" and "Ubuntu 12.10 2.vbox-prev". There was no "Ubuntu 12.10 2.vdi" file as the "Ubuntu 12.10.vdi" file was re-used for this VM and remained in the "C:\Users\Test\Desktop\ubuntu-12.10-desktop-i386\" directory. A recovery file was generated, "Ubuntu 12.10 2.vbox-tmp", in the "C:\Users\Test\VirtualBox VMs\Ubuntu 12.10 2" directory. A log directory was created after the VM was booted; the WhatChanged logs showed that there were two log files, "VBox.log" and "VBox.log.1", indicating that the VM was booted twice. Before any user activity was generated

on this VM, a snapshot was taken of the running state. This was documented in the change management logs; the "C:\Users\Test\VirtualBox VMs\Ubuntu 12.10 2\Snapshots" entry documented the creation of a folder for the snapshot files. The logs document a file with the extension of .sav was present in the Snapshots directory. The SAV file is the snapshot configuration file that contains the complete state of the guest at a certain point in time (Mehnert, 2012). User activity generated after the snapshot was not documented in the change management logs. After the user activity test, the Ubuntu 12.10 2 VM was reverted to the snapshot. The FolderChangesView documented that the "ca267327-7a44-4a85-b116-443fb31d2fe0}.vdi" file was created in the "C:\Users\Test\VirtualBox VMs\Ubuntu 12.10 2\Snapshots" directory.

The Ubuntu 13.04 VM was designated for the VM deletion test. After the "Delete all files" option was selected in VirtualBox Manager, the change management logs show that all the files, as well as the directory "C:\Users\Test\VirtualBox VMs\Ubuntu 13.04" were deleted. This included all the log files. The analysis of the hard drive would confirm if these files were no longer present or recoverable.

The Regshot, FolderChangesView and WhatChanged log files provided evidence documented in the registry keys and file directory locations of the test machine. This information was used to identify areas that would be analyzed during the examination of the RAM and hard drive image. While an examiner would not normally have this type of documentation to assist in the analysis, these log files were useful in pinpointing potential locations of evidence during the forensic analysis of the RAM and hard drive.

Forensic Analysis

Forensic Toolkit® (FTK) v1.81.5 was used to conduct the forensic analysis of the disk image files and the memory image file. FTK is a digital investigations platform that is considered

an industry standard and court-accepted. It has the capability to create images, analyzes files and file structures, and carve data from unallocated space (AccessData, 2013). FTK v1.81.5 is a demonstration version; a maximum of 5000 files can be examined in one case. The E01 file was too large to be added to one case and therefore it was required to split the file into two sections for effective analysis. FTK Imager was used to export the satellite_disk_image.E01 file into three image files; Partition 1, Partition 2 and Unallocated Space. A case was created for each of these E01 files, as well as one for the memory dump file. These FTK case files were the primary focus of the analysis.

The analysis strategy consisted of three goals; confirmation of the data provided by the change management logs, the recovery of any user-generated data from the VMs, and the recovery and restoration of any VMs. The analysis began with the memory dump file and progressed through the image files. The goal was to analyze the registry files and user files for evidence of the installation of VirtualBox and any VMs, examine the RAM and unallocated space for evidence of the user-generated browser and file activity conducted in the three VMs, and search for the VBOX and VDI files required to restore a VM for analysis.

Memory forensics. The "Memdump.mem" image file was imported into FTK, and a new case was created. The search terms "virtualbox", "oracle", "vbox", and "vdi" were entered into the indexed search function. This search returned 4,127 hits for "virtualbox". The results were copied to a text file for review. The review of the search results identified multiple locations in Program Files, User files and the registry where the VirtualBox application files were stored, more than can be listed here. Suffice to say that there was a multitude of evidence available in the RAM illustrating that VirtualBox was indeed installed on the machine and showed the user account under which it was installed.

A search for keywords was conducted to find evidence of the VMs that were installed. The search results showed evidence of three VMs that had been installed on the system: Ubuntu 13.04, Ubuntu 12.10 and Ubuntu 12.10 2. The “oracle” search returned 2,239 hits. The results were similar to the results for “virtualbox”; a multitude of Program File, User file, and registry locations revealing that VirtualBox had been installed in Oracle directories (Programs/Oracle, Program Files/Oracle, Start Menu/Oracle, etc.), as VirtualBox is an Oracle product. The “vbox” search returned 2,022 hits. A review of these results showed “vbox” as not only a file extension associated with VirtualBox operating files, but also as part of the directories and registry keys associated with the application. The “vdi” search returned far fewer hits; only 207. The review of the search results showed that the “vdi” references were associated with file extensions only, except in the VirtualBox setting directory that identifies all the file types that VirtualBox is compatible with and can use to build a VM.

A text file was created which contained search terms from the user-generated activity. These included the names of the JPG files, PDF documents, ODT files, Google searches and URLs. This text file was imported into the indexed search function in FTK and used to find evidence in the drive free space. This search returned 6 hits for “secureflorida”. In the Ubuntu 12.10 VM test, the analyst had navigated to www.secureflorida.org/legalissues/computer_laws/. During the search of the RAM, the URL “http:// www.secureflorida.org/legalissues/computer_laws/” was found in a portion of the drive free space. The search hit also showed Mozilla Firefox as the web browser; Firefox was not installed on the host system but was installed on the guest VMs. Hacker10 was the only other search result related to the user-generated activity in the VMs, specifically during the Ubuntu 12.10 test. The URL http://www.hacker10.com/ was revealed in 6 hits from the search. Included in two of the

results was a host IP address which matched one of the IPs listed in the Network ARP information harvested through the volatile data acquisition. All other search keywords related to the user-generated activity turned up no hits.

The change management logs provided indications of the locations where the VirtualBox application created files during installation and during the creation of the VMs. A checklist was created in order to compare the RAM search results against the change management logs. Most, but not all, of the evidence provided by the log files could be connected to findings in the search output. The items not found in the RAM were documented in order to add to the search of the hard drive image files.

Hard drive image forensics. As was noted, the E01 image file was too large to attach to one FTK 1.81.5 case and was split into three files; Partition 1, Partition 2 and Unallocated Space. Each of these images was analyzed in separate FTK cases. The analysis strategy consisted of conducting a string search of each partition, retrieval and analysis of the registry files, retrieval and analysis of the pagefile.sys file, and the analysis of the C:\Users\Test directories.

Keyword search. The Partition 1 Image case was opened in FTK. Under the Explore tab, the directory tree was extended so that all sections could be viewed. A quick overview of the directories was conducted to assess where evidence might be located. The search terms text file was imported into the Indexed Search. The results returned only hit on the keywords “vbox”, “vdi” and “oracle”. A comparison of the search results to the evidence checklist revealed no new evidence than was discovered during the RAM analysis. The investigation proceeded to the Partition 2 Image case file.

The Partition 2 Image case was opened in FTK and the search terms text file was imported into the Indexed Search. The results returned hits for the VirtualBox files search terms (vbox,

virtualbox, vdi, and oracle) and some of the user-generated activity search terms. These results were exported to a Microsoft Excel file for analysis. The search results were used to identify potential evidence files.

The analysis of the search results show evidence that the Mozilla Firefox web browser was used for browsing on the Ubuntu 12.10 and Ubuntu 12.10 2 VMs, as Firefox is the default browser in these Ubuntu OSs. The results showed that the "61aUpZ7V1qL._SL1499_.jpg" image file and the "9195-main.jpg" image file were downloaded in the Ubuntu 12.02 VM. These images were results of the Google search for "girl halloween costumes" that was conducted on the Ubuntu 12.10 machine. Evidence of that Google search was also present in the indexed search results. The Google search "girl halloween costumes.pdf" that was conducted on the Ubuntu 12.10 VM was evident in the indexed search file, as well as the URL http://www.epilogsys.com/ScoutingWeb/Documents/Silver%20Proj%20Ideas.pdf that was selected from the Google search results. Evidence of the document "Silver Project Ideas.pdf" that was downloaded from this website was present in the indexed search results.

The indexed search showed that the "girl-vampire-costume.jpg" file was downloaded in the Ubuntu 12.10 2 VM. Evidence of the Google search for "Halloween costumes.pdf" was found in the indexed search results for Ubuntu 12.10 2. The Google result selected from this search was http://lmc.gatech.edu/~cpearce3/PearcePubs/LudicaDress-UP.pdf. Evidence of this URL as well as the "LudicaDress-Up.pdf" that was downloaded from the website was present in the indexed search results.

The "halloween.odt" file that was created in the Ubuntu 12.10 VM was listed in the indexed search results. It was listed as present in the Ubuntu 12.10 2\Snapshots section of the partition under the home/Ubuntu/Documents directory. The results also show that

“halloween.odt” is a LibreOffice Writer document. The “halloween1.odt” file that was created in the Ubuntu 12.10 2 VM was listed in the indexed search results and that it was a LibreOffice Writer document.

The URL http://www.secureflorida.org/legalissues/computer_laws/computer_laws/ was selected in Ubuntu 12.10 2 from the Google search for “cyber crime laws florida”. The URL was listed in the indexed search results, but the Google search was not. The URL www.hacker10.com was found in the indexed search results. No evidence of the user-generated activity from the Ubuntu 13.04 VM was present in the indexed search results.

The unallocated space FTK case was analyzed for evidence. The unallocated space file contained 4 file items, but no evidence was found in the slack space. A search was performed for JPG, PDF and ODT file headers. None of these were found. The analyst proceeded to the file recovery phase.

Registry analysis. The analyst added the “satellite_disk_image.E01” image file to FTK Imager. The analyst navigated to the Windows\System32\config directory and exported the SAM, SECURITY, SOFTWARE and SYSTEM registry hives to the 250 GB external hard drive. An MD5 hash was conducted for each file to ensure file integrity after the analysis.

AccessData Registry Viewer version 1.6.3.34 was used to analyze the registry files that were identified as evidence in the change logs. Registry Viewer is an application that is used to view Windows system registry files from any source (AccessData Corp., 2007). The analyst opened the Software registry hive and navigated to SOFTWARE\Classes. This location was identified in the change logs as evidence of the existence of the VirtualBox application installed on the test machine. Figure 12 displays those associated with VirtualBox.

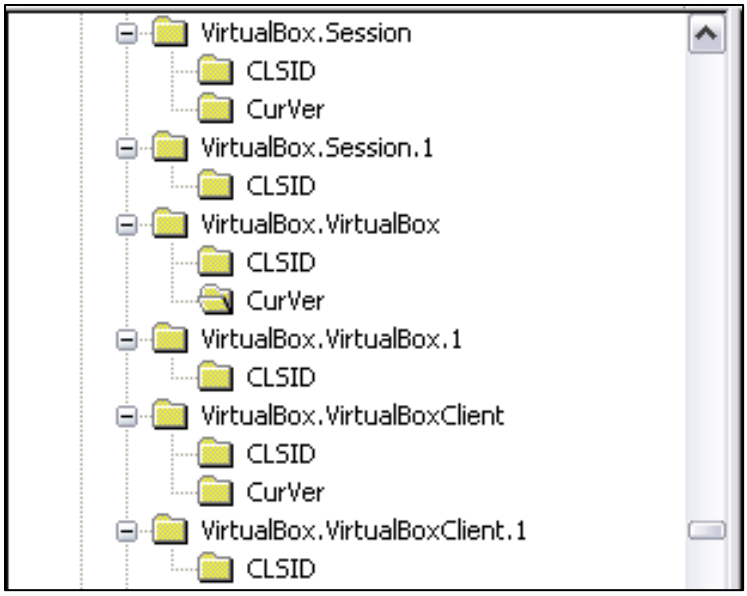
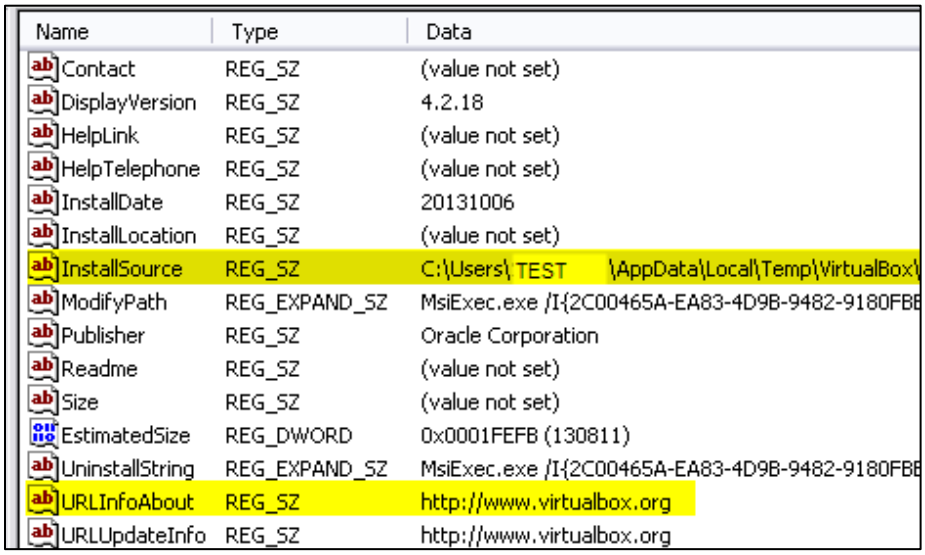


Figure 12. SOFTWARE registry keys. There are several associated with VirtualBox.

The "SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\A56400C238AEB9D449281908BFBE4DCA\InstallProperties\URLInfoAbout: http://www.virtualbox.org" registry key documented in the change logs was identified. Figure 13 shows evidence of where the VirtualBox application was obtained.



Name	Type	Data
Contact	REG_SZ	(value not set)
DisplayVersion	REG_SZ	4.2.18
HelpLink	REG_SZ	(value not set)
HelpTelephone	REG_SZ	(value not set)
InstallDate	REG_SZ	20131006
InstallLocation	REG_SZ	(value not set)
InstallSource	REG_SZ	C:\Users\ TEST \AppData\Local\Temp\VirtualBox\
ModifyPath	REG_EXPAND_SZ	MsiExec.exe /I{2C00465A-EA83-4D9B-9482-9180FBE
Publisher	REG_SZ	Oracle Corporation
Readme	REG_SZ	(value not set)
Size	REG_SZ	(value not set)
EstimatedSize	REG_DWORD	0x0001FEFB (130811)
UninstallString	REG_EXPAND_SZ	MsiExec.exe /I{2C00465A-EA83-4D9B-9482-9180FBE
URLInfoAbout	REG_SZ	http://www.virtualbox.org
URLUpdateInfo	REG_SZ	http://www.virtualbox.org

Figure 13. SOFTWARE registry keys.

The analyst opened the System registry hive, navigated to "HKLM\SYSTEM\ControlSet001\Control\Session

Manager\Environment\VBOX_INSTALL_PATH: “C:\Program Files\Oracle\VirtualBox\” and documented the install path. Figure 14 shows the VBOX_INSTALL_PATH where the application was installed.

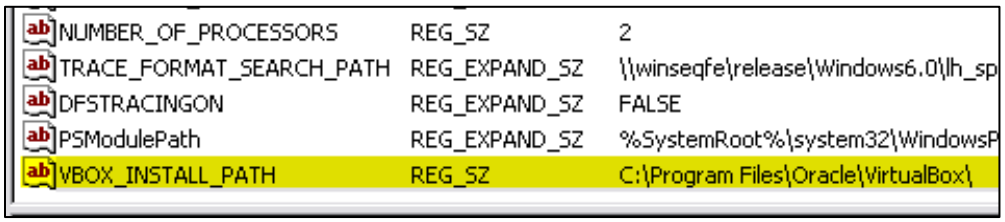


Figure 14. SYSTEM registry keys.

The “HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall” registry key contained evidence showing the version of VirtualBox that was installed as Oracle VM VirtualBox 4.2.18. Figure 15 shows that the install date of the VirtualBox application was located in the same registry key and was validated against the date that was recorded during testing.

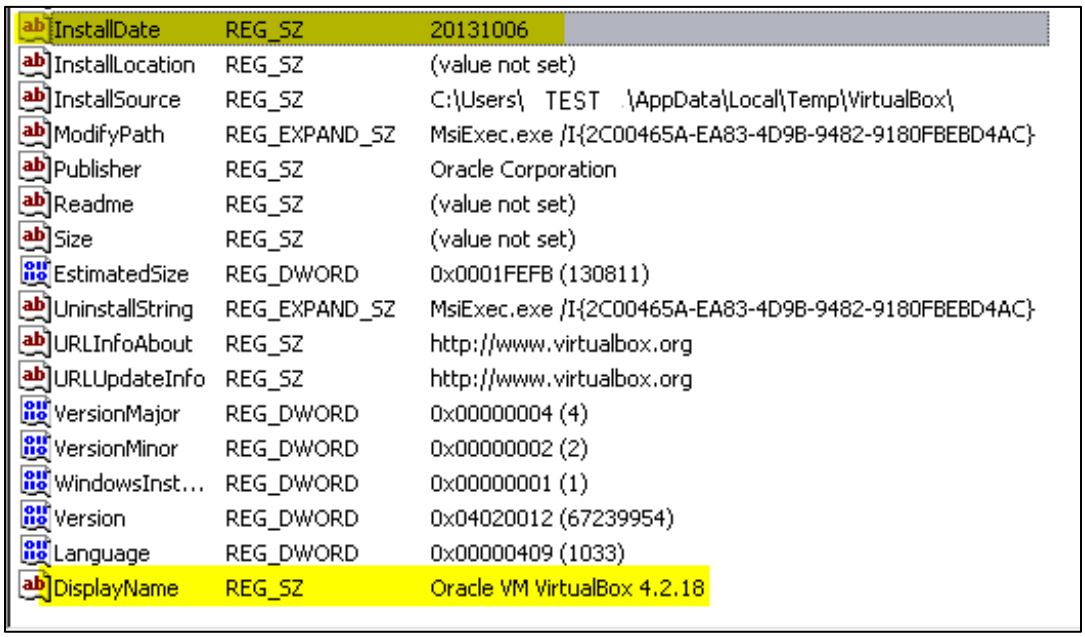


Figure 15. SOFTWARE registry keys.

The “SYSTEM\ControlSet001\Control\DeviceClasses\{cac88484-7515-4c03-82e6-71a87abac361}” registry key showed evidence of VirtualBox; VirtualBox was originally a

product of SUN before it was an Oracle product. Figure 16 shows the entries containing SUN, not Oracle.

Figure 16. SYSTEM registry keys.

The analysis of the registry files confirmed the locations of registry key evidence that was documented in the change management logs. These registry entries show the various locations in the registry where the VirtualBox application created files during installation. The remainder of the VirtualBox files was documented in the root (C:) directory of the hard drive image.

Root directory. The root directory containing the bulk of the files and folders used by the system and the user files was located on Partition 2 of the "satellite_disk_image.E01" image file. Due to the restrictions on the number of files that can be examined imposed by the demo version of FTK, the image file was examined using FTK Imager. In FTK Imager, the file structure could be explored and the contents of the folders examined. In this way, the locations of evidence indicated in the change management logs were confirmed. Several locations of evidence were identified in the change log files, primarily in the C:\Windows\Prefetch, C:\Windows\System32\DRVSTORE, and C:\Users\Test locations. Evidence that the VirtualBox

application was run was found in the Windows\Prefetch files. Figure 17 shows the prefetch files as evidence that the application was run on the system.

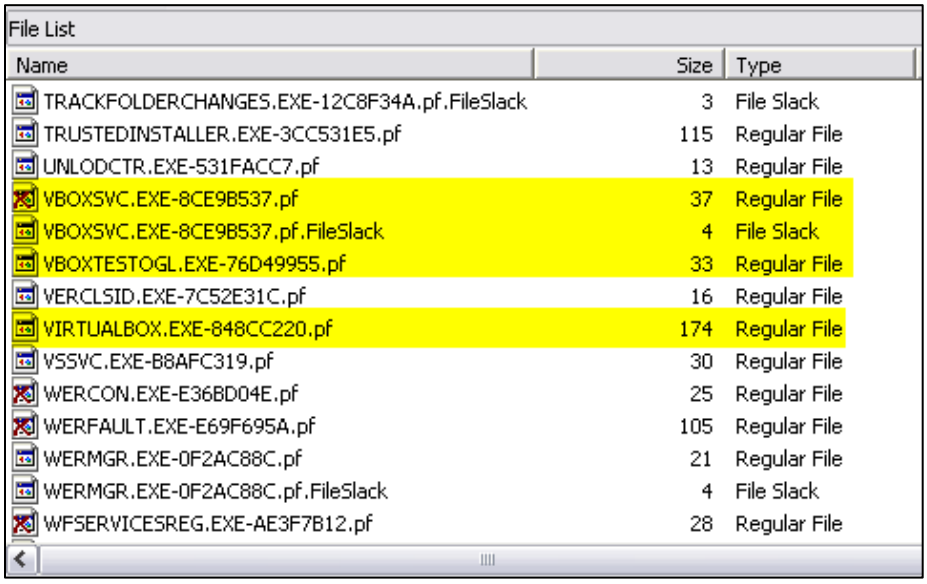


Figure 17. Prefetch files.

The Windows\System32\DRVSTORE location contained two directories associated with the VirtualBox application. Figure 18 shows this location and the files.

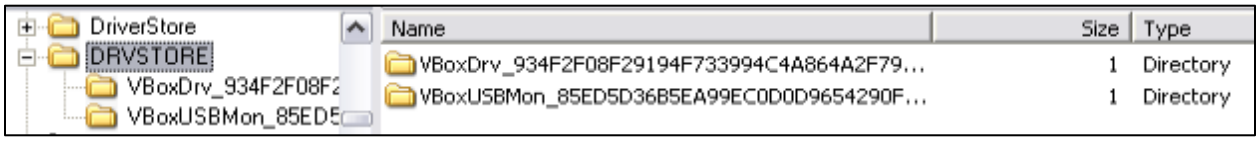


Figure 18. Evidence of VirtualBox in the System32 directory.

The root\Program Files directory and the root\Users\Test directory contained the most evidence; this was expected, as this is where the VirtualBox application files and VM files are stored by default. The analyst navigated to root\Program Files\Oracle\VirtualBox and exported the VirtualBox folder to the external hard drive. The analyst then navigated to root\Users\Test\VirtualBox VMs and exported the VirtualBox VMs folder. The \Virtualbox VMs folder contained the VirtualBox files for all three VMs that were installed, including the deleted Ubuntu 13.04. The Ubuntu 13.04 folder icon has a red X on it, signifying that it had been deleted. Figure 19 illustrates the existence of the three VMs that were created on the test system.

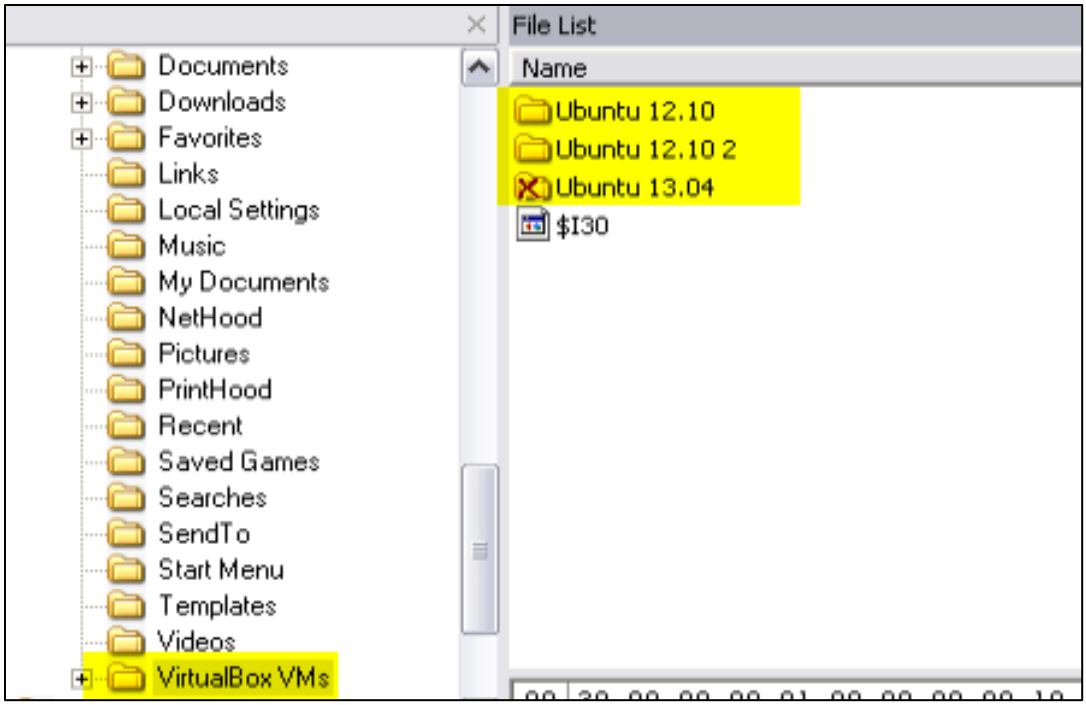


Figure 19. The User directory showing three VMs installed.

The analyst navigated to root\Users\Test\Desktop\ubuntu-12.10-desktop-i386.7z and exported the folder to the external hard drive. The final folder containing the VirtualBox application files, root\Users\Test\.VirtualBox, was exported to the external hard drive. These files would be used to attempt to reconstruct the VMs and examine them.

The analyst examined the exported folders in order to confirm the findings of the change management log files during testing. The VirtualBox VMs folder contained the folders of all three VMs. The "ubuntu-12.10-desktop-i386.7z" file contained the "ubuntu-12.10-desktop-i386.vbox" and "ubuntu-12.10-desktop-i386.vdi" files identified in the change management logs. The Ubuntu 12.10 VM folder contained the Logs folder, "Ubuntu 12.10.vbox" and the "Ubuntu 12.10.vbox-prev" files that were identified in the change management logs. The three "VBox.log" files that were documented in the FolderChangesView logs were located in the Logs folder.

The process of exploring the root directory confirmed all the findings documented in the change management logs. This process also facilitated the identification and extraction of evidence files, namely the VirtualBox application files and the VMs. An attempt would be made to restore the VMs in order to retrieve evidence.

Pagefile.sys analysis. The analyst exported the file using FTK Imager. The analyst imported the exported pagefile.sys file into FTK for analysis and examined the file for PDF and JPG files using the data-carving feature. No evidence of the user-generated activity was found in the carved files. The analyst imported the search keywords text file into the Indexed Search function. No evidence was found in this search of the files.

VM recovery. The analyst exported the recovered VMs that were installed on the Test system to the 250 GB external hard drive. MD5 hashes were generated for each of the VBOX and VDI files before any actions were taken. The analyst attempted to recover the Ubuntu 12.10 VM. The folder contained the intact log files, the VBOX file and the VDI files. The analyst launched the VBOX file using VirtualBox, but an error was received; "Failed to open the hard disk …. Cannot register the hard disk C:\path\to\new\vdi with UUID {xxxx} because a hard disk C:\path\to\old\vdi already exists in the media registry (C:\path to VirtualBox.xml)". This indicated that the UUID associated with the Ubuntu 12.10 VM was already registered. A UUID is a unique identifier that VirtualBox uses to identify virtual machines. VirtualBox assigns a (UUID) to each disk image to make sure it is only used once, which precludes the VDI from being cloned by merely copying it (Oracle Corporation, 2013). In order to work around this dilemma, the analyst downloaded a free open-source application called CloneVDI. CloneVDI is a tool that can be used to assign a new UUID to a cloned VDI file. It was developed by MPack, a moderator on the VirtualBox.org forum (MPack, 2009). The analyst downloaded CloneVDI

version 2.10 and used it to clone the "ubuntu-12.10-desktop-i386.vdi" file (MPack, 2009). The cloned VDI file did not have the same MD5 hash as the original, as the UUID was changed. The analyst used this VDI file to build a new VM in VirtualBox titled "Ubuntu 12.10 Cloned". The analyst launched and booted the VM successfully. The analyst recovered the two JPG images that were downloaded, the PDF document that was downloaded, the browser history and the LibreOffice ODT file that was created during the testing phase of the Ubuntu 12.10 VM.

The Ubuntu 12.10 2 VM was used to test the recovery of a VM reverted to a snapshot. The recovered files included two VDI snapshot files and the VBOX file associated with the Ubuntu 12.10 VM. The original VDI file that had been used to build the VM was the same as was used for the Ubuntu 12.10 VM. The analyst attempted to boot the Ubuntu 12.10 2 VM using the cloned "ubuntu-12.10-desktop-i386.vdi" file as the base image, with the hopes that the VM would recognize and load the snapshot files. This failed, as the recovered virtualbox.xml file, the file that contained the pointers for the snapshots, specified the original test machine file path.

The recovered Ubuntu 13.04 VM files contained the log files, the VBOX file and the VDI file. The VM could not be built, because the VDI file was a zero-byte file containing no data. The deletion of the VM files left the VDI file and VBOX file, but removed the data on the virtual disk drive rendering it unrecoverable. No snapshots or backups of the VDI file were created to restore the VM.

File carving. The RAM and hard drive image partitions were analyzed for recoverable evidence files pertaining to the user-generated activity that was conducted on the three VMs. The earlier keyword search conducted on the RAM image yielded a wealth of references to the VirtualBox application files, the VM files and the activity conducted inside the VMs. The analyst attempted to recover the web history, images and documents created on the VMs during testing.

The analyst launched the RAM image case file in FTK and used the Data Carve tool to parse the image for evidence files. While a number of files were retrieved, no pertinent evidence was discovered

The analyst launched the Partition 1 image case file in FTK and used the Data Carve tool to parse the image for evidence files. Again, no pertinent evidence was discovered. The same process was used for the Partition 2 image, with the same results. The analysis of the Unallocated space image in FTK using the Data Carving tool yield negative results.

Although the recovered snapshot VDI files from the Ubuntu 12.10 2 VM were unable to be loaded into VirtualBox to create a functioning VM, there was a chance that evidence could still be recovered from them. In an effort to ascertain the validity of this theory, the analyst loaded each of the two snapshot VDI image files into FTK for analysis. The analyst used the Data Carving tool against each, with positive results. Evidence of the user-generated activity conducted on the Ubuntu 12.10 2 VM before it was reverted to the snapshot was recovered. The "girl-vampire-costume.jpg" was recovered and opened in the native JPG format. The evidence is pictured in Figure 20.

Figure 20. girl-vampire-costume.jpg

A thumbnail image showing the Google search "halloween costumes .pdf" and the selected results “[PDF] Playing Dress-Up: Costumes, roleplay and imagination” was recovered. The evidence is pictured in Figure 21. This was a recovered thumbnail and is difficult to view, but the google search and the results can be read.

Figure 21. Thumbnail: halloween costumes .pdf Google Search results.

A thumbnail showing a portion of the PDF downloaded from http://lmc.gatech.edu/~cpearce3/PearcePubs/LudicaDress-Up.pdf was recovered. The thumbnail was very difficult to read, but knowing the reference helped to identify it. The enhanced evidence is pictured in Figure 22.

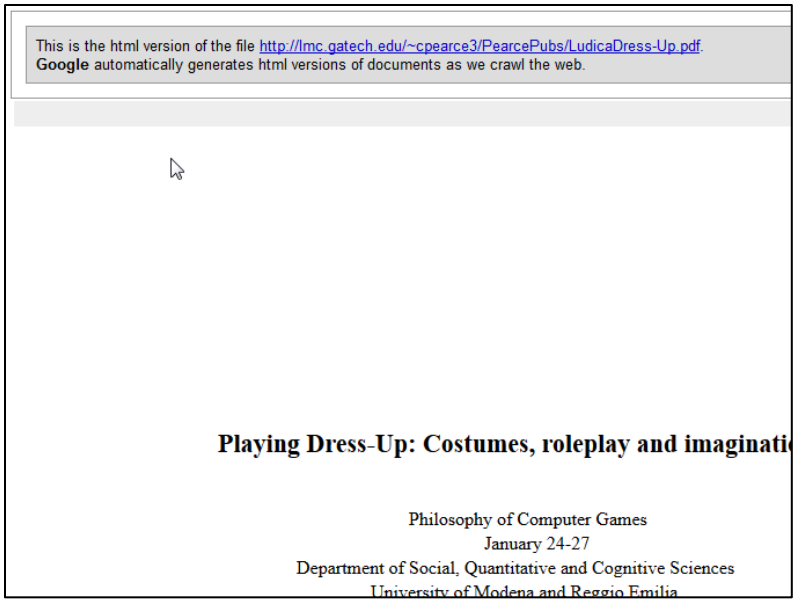


Figure 22. Thumbnail: LudicaDress-Up.pdf downloaded.

A thumbnail of the “halloween 1.odt” LibreOffice Writer file with the words "I love candy" was recovered. The thumbnail was very difficult to read, but knowing the reference helped to identify it. The enhanced evidence is pictured in Figure 23.

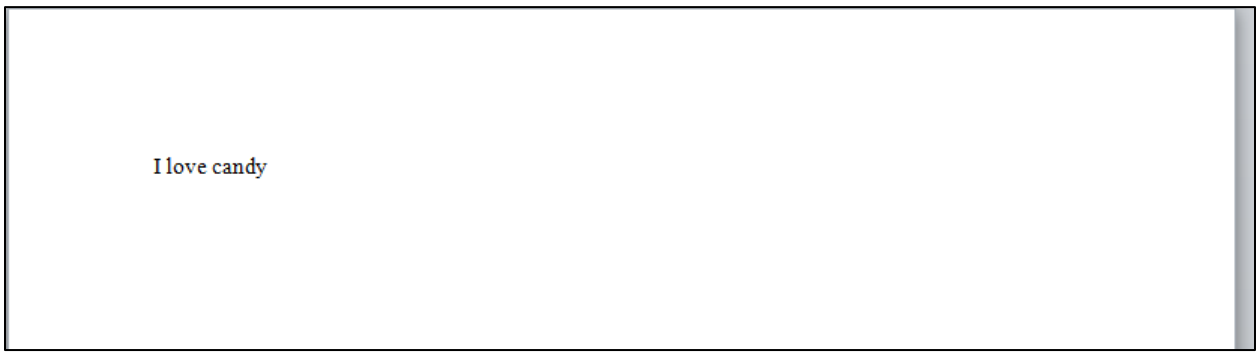


Figure 23. Thumbnail: halloween 1.odt.

The recovery of this evidence from the snapshot VDI files was a promising discovery. The same attempt could not be made to recover evidence from the deleted Ubuntu 13.04 VDI file, as it contained no data.

Discussion of Findings

The analysis of the data collected after the testing phase sought to find evidence validating the installation of the VirtualBox application on the test machine, confirming the

creation of the three VMs and finding any evidence of the user activity that was generated inside the VM environments. The Regshot, FolderChangesView and WhatChanged logs that were generated during the testing phase were used as the starting point of the analysis. Once the documentation was reviewed, the goal was to conduct a forensic analysis and validate the documentation, since the documentation was not considered part of the evidence. The other objectives of the forensic analysis were to recover evidence of the user-generated activity and to possibly recover and restore one or more of the VMs.

The registry and file system changes that were created during each phase of testing provided insight into the installation and operation of the VirtualBox application and the VMs. The change management logs showed evidence of the installation of VirtualBox in the SOFTWARE and SYSTEM registry files as well as in the C:\Program Files, C:\ProgramData, C:\Windows\system32, C:\Windows\Prefetch and C:\Users directories. These locations were documented so that they could be validated; these log files do not constitute evidence as they would normally not be available.

The review of the change management logs and the subsequent forensic analysis of the RAM and hard drive images revealed evidence of the existence of the VirtualBox application, the existence of all three VMs, and evidence of the user-generated activity. The objective in recovering evidence of the installation of the VirtualBox application was two-fold; the significance of evidence of virtualization technology to the case, and an overall inspection of the VirtualBox file system and its interaction with the host machine. The existence of a virtualization application is significant to a forensic investigation as it indicates that there may be evidence located in areas other than the host system itself. As was previously discussed, the use of virtualization technology does not necessarily indicate criminal activity; the benefits of

virtualization make it an effective solution for many legitimate business and personal deployments. A forensic analyst will need to know if there are VMs that could be recovered and analyzed for evidence and what role the existence of VMs play in the overall case. An understanding of the types and locations of VirtualBox application files and the changes made to the host OS would also be of importance for a forensic analyst. These principles were applied in the analysis.

The analysis of the RAM was conducted by using the Indexed Search function in FTK to search for keywords gleaned from the change management logs and those documented from the user-generated activity. These terms were added to text files and imported into FTK. The VirtualBox-specific keywords returned a multitude of results that matched those documented in the change management logs. The search results confirmed that three VMs were installed on the system; Ubuntu 12.10, Ubuntu 12.10 2, and Ubuntu 13.04. The user-generated activity keywords returned results that showed two of the URLs that were navigated to during testing; the www.secureflorida.org and www.hacker10.org, which were navigated to on the Ubuntu 12.10 and Ubuntu 12.10 2 VMs. The results showed that Mozilla Firefox was the browser used to navigate to these URLs. A host IP address which matched one of the IPs listed in the Network ARP information harvested through the volatile data acquisition was also returned in the search results. The existence of the URLs in the RAM illustrate that there is at least some network activity that escapes the confines of the VM.

The analysis of the hard drive image was accomplished through several avenues; a keyword search, an analysis of the registry files, exploring the root directory, analyzing the pagefile.sys file and attempting to recover and reconstruct the VMs. The same VirtualBox-specific and user-generated activity keywords were used to identify any evidence. Again, there

were many results for the VirtualBox-specific files. The more interesting results were those that were returned against the user-generated activity keywords. The results showed that the "61aUpZ7V1qL._SL1499_.jpg" image file and the "9195-main.jpg" image file were downloaded in the Ubuntu 12.10 VM. The Google search "girl halloween costumes.pdf" that was conducted on the Ubuntu 12.10 VM was evident in the indexed search file, as well as the URL http://www.epilogsys.com/ScoutingWeb/Documents/Silver%20Proj%20Ideas.pdf that was selected from the Google search results. Evidence of the document "Silver Project Ideas.pdf" that was downloaded from this website was present in the indexed search results. The keyword search returned results of the "girl-vampire-costume.jpg" file that was downloaded in the Ubuntu 12.10 2 VM as well as the Google search term "Halloween costumes.pdf" conducted on the same machine. The Google result selected from this search was http://lmc.gatech.edu/~cpearce3/PearcePubs/LudicaDress-UP.pdf, which was returned in the results as well as the LudicaDress-Up.pdf that was downloaded from the website. The "halloween.odt" file that was created in the Ubuntu 12.10 VM was listed in the indexed search results, and the file path of the file was also listed in the results. The indexed search results also show that "halloween.odt" that was created in the Ubuntu 12.10 2 VM was classified as a LibreOffice Writer document. The URL http://www.secureflorida.org/legalissues/computer_laws/computer_laws/ was selected in Ubuntu 12.10 2 from the Google search for "cyber crime laws florida". The URL www.hacker10.com was found in the indexed search results as well. During this search, no evidence of the user-generated activity from the Ubuntu 13.04 VM was found.

The registry analysis was successful in confirming the multiple locations where the VirtualBox application files could be found on the system. The registry hives were extracted

from the hard drive image file using FTK Imager and analyzed using AccessData's Registry Viewer. All of the registry files listed in the change management logs were validated against the registry files in the SYSTEM and SOFTWARE registry keys. These results provided evidence of where the application was downloaded from, when it was downloaded, which user downloaded it, and where it was installed.

The analysis of the root directory on the hard drive image provided confirmation of the change log file documentation. The C:\Windows\Prefetch, C:\Windows\System32\DRVSTORE, and C:\Users\Test locations all contained the evidence that was observed in the log files. The main goal in analyzing the root directory was to recover the VirtualBox VMs that had been created. The VirtualBox folder in the root\Program Files\Oracle directory was exported to the external hard drive. This is the main directory to which the VirtualBox application files were installed. The VirtualBox VMs folder was located and exported to the external hard drive. This folder contained the files for all three VMs. The ubuntu-12.10-desktop-i386.7z folder was located in the root\Users\Test\Desktop directory and exported to the external hard drive. This folder contained the "ubuntu-12.10-desktop-i386.vdi" file that would be needed to reconstruct the Ubuntu 12.10 and Ubuntu 12.10 2 VMs.

The attempt to recover the Ubuntu 12.10 VM initially failed; the UUID associated with the Ubuntu 12.10 VM was already registered to the test system and VirtualBox would not allow a new VM to be created using the "ubuntu-12.10-desktop-i386.vdi" file. CloneVDI was used to assign a new UUID to a cloned VDI. This cloned VDI was used to build the new VM, and the recovered Ubuntu 12.10 VM was successfully booted. The analysis of the VM recovered the two JPG images that were downloaded, the PDF document that was downloaded, the browser history that was created and the LibreOffice ODT file that was created during the testing phase. The

attempt to restore the Ubuntu 12.10 2 VM was unsuccessful because the recovered virtualbox.xml file, the file that contained the pointers for the snapshots, specified the original test machine file path. In order to make this work, the xml file would have to have been altered to point to the new file path. This attempt was made, but was still unsuccessful. The restoration of the Ubuntu 13.04 VM was unsuccessful due to the VDI file having no data whatsoever. The recovery of the VMs was not a complete loss, as it was demonstrated that a VM can be recovered and restored, and evidence recovered from it as was accomplished with the Ubuntu 12.10 VM which was completely restored.

Although the recovered Ubuntu 12.10 2 snapshot files could not be used to restore the VM, evidence was able to be recovered from them. The two snapshot VDI files were loaded into FTK, and the Data Carving tool was used to recover evidence of the user-generated activity that was conducted on the VM before it was reverted to the snapshot. The “girl-vampire-costume.jpg” was recovered and opened in the native JPG format. A thumbnail image showing the Google search "halloween costumes .pdf" and the selected results “[PDF] Playing Dress-Up: Costumes, roleplay and imagination” was recovered. A thumbnail showing a portion of the PDF downloaded from http://lmc.gatech.edu/~cpearce3/PearcePubs/LudicaDress-Up.pdf was recovered. A thumbnail of the “halloween 1.odt” LibreOffice Writer file containing the text "I love candy" was recovered. The recovery of this evidence from the snapshot VDI files was a promising discovery. The same attempt could not be made to recover evidence from the deleted Ubuntu 13.04 VDI file, as it contained no data.

Overall, the test and analysis of the VirtualBox VMs was deemed a success. The recovery of evidence of the user-generated activity from within the VMs demonstrates that there is some transference of data from the guest system; the Ubuntu 12.10 2 snapshot image files yielded

evidence when loaded into FTK and analyzed. The recovery of evidence from the image files was not surprising; what was unexpected was that the indexed search of the RAM revealed traces of the user generated activity on two of the three VMs, especially from the reverted VM. The recovery and restoration of the Ubuntu 12.10 VM was a successful exercise that demonstrated the UUID problem and how it was resolved in the successful cloning of a VDI. The evidence that was recovered from the Ubuntu 12.10 2 VM snapshot images of files that were not present in the VM after it was reverted to the snapshot during testing showed that some evidence remains after the rollback to a snapshot. The failure to restore the deleted Ubuntu 13.04 VM confirms that deleted a VM is an effective way to destroy evidence, demonstrating another avenue for cybercrime to flourish.

Future Research Recommendations

The purpose of this study was to examine the possibility of recovering forensic evidence of user activity within an Oracle VirtualBox virtual machine (VM) that had been deleted or reverted to a restored point. Virtualization technology was introduced and the benefits and limitations of virtualization were discussed. Virtualization is a relatively inexpensive and effective way to create test environments and to expand the infrastructure of an organization while decreasing the amount of hardware required. Virtualization can also be an effective way to obfuscate evidence of cybercrime, making it a point of interest in the digital forensic community.

The focus of this study was on the use of Oracle VirtualBox, a free open-source virtualization application. Linux Ubuntu versions 12.10 and 13.04 were used as the guest OSs. These are also free open-source software available for download from the internet. The combination of these applications created a free virtualization solution that could be used to hide or destroy evidence of criminal activity. The commonly held notion is that any activity

conducted in a VM stays in the VM and does not transfer to the host machine; that web browsing activity, chat, email and files all stay inside the VM. If that VM is deleted, or a clean snapshot was created that the VM could be reverted to after criminal activity is conducted, evidence is effectively removed. This study attempted to ascertain the validity of this assumption.

The testing phase was performed to collect data on the impact that the installation of VirtualBox had on the test system, in an effort to study and document the file structure of the application. The testing phase also included the creation of three VMs; one for removal from the VirtualBox manager, one to revert back to a snapshot, and one for deletion. Data was collected during each phase of testing in the form of change management monitoring applications. After the tests were performed, a forensic acquisition of the volatile data, RAM and the hard drive was accomplished.

The analysis phase sifted through the collected data in an effort to discover any evidence of the user-generated activity and to attempt to recover and restore the VMs. The change management log files generated throughout the testing process using the Regshot, FolderChangesView and WhatChanged applications were used as a starting point in the analysis. The RAM image and hard drive image were analyzed with the objective of recovering evidence of the installation of Oracle VirtualBox. The existence of VirtualBox on a machine seized as evidence is not necessarily an indication of criminal activity, but it would alert the forensic analyst that there may be evidence located on VMs installed on the machine. The forensic analysis was also conducted with the objective of recovering evidence of activity generated by the user in the VMs and possibly recovering the VMs themselves for analysis. Several tools and techniques were used in the analysis; indexed searches using keywords, traversing the file

structure, exporting files and carving files from unallocated space. AccessData's FTK, FTK Imager and Registry Viewer applications were used for the analysis.

The indexed searches were used as a starting point; a way to validate the change log findings and serve as a quick way to see if there was any promising evidence. The file traversal provided a view into the file structure and folder contents. Promising files were exported for further analysis. Data carving was used to recover files that no longer existed in the user's files, but may exist in unallocated space.

The analysis was successful in recovering the VM files and restoring one VM. The Ubuntu 12.10 VM was successfully recovered and restored, allowing the analyst the ability to navigate inside the OS and discover evidence of the user-generated activity. The Ubuntu 12.10 2 VM could not be recovered, as the attempt to rebuild it using the exported files failed due to incompatible specifications in the XML file. The analysis was successful in recovering evidence of the user-generated activity. The browsing history, files downloaded and files created on the Ubuntu 12.10 VM were referenced in the indexed search, and were able to be recovered directly from the restored VM. Some browsing history, the downloaded JPG file, evidence of the downloaded PDF file and evidence of the LibreOffice ODT file that were generated in the Ubuntu 12.10 2 VM prior to it being reverted back to the saved snapshot were recovered. The evidence was recovered from the snapshot VDI image files; the VDI files could not successfully be used to restore the Ubuntu 12.10 2 VM, but were successfully loaded into FTK and analyzed like any other image file. The evidence recovered from the snapshot files provided insight into the user-generated activity conducted in the VM prior to it being reverted to a snapshot. Unfortunately, the Ubuntu 13.04 VM was successfully restored; although the VBOX and VDI

files were exported, the VDI file was a zero-byte file containing no data. This proved that a deleting a VM is an effective way to destroy evidence.

The final objective of this research was to determine if there is a cost benefit of allocating the necessary resources to perform the associated tasks when strategizing a plan for forensic analysis. The analysis was conducted using common forensic applications that would most likely be available in a digital forensic lab or freeware that can be easily downloaded at no cost. The effort to recover the evidence was no more than would be expended during the course of an average investigation; searches, examining file structures, exporting files and performing data carving are all common activities during an investigation. The restoration of the Ubuntu 12.10 VM required only that the forensic analyst properly clone the VDI file and know how to use VirtualBox to build a VM. Overall, the process was no more time-consuming, costly or requiring specialized knowledge than would normally be expected.

There are several virtualization solutions available. Future work on this subject should include the testing of a different virtualization application; VMWare, Parallels and XenServer are other options. Other OSs can be used as guests on the VM; there are many other Linux OSs, Windows and Mac OSs that can be installed as VMs. These should be tested to confirm or disprove the findings that the guest OSs can transfer data to the host OS. The findings should also be analyzed for causality; this study did not attempt to discover the reason for the data transfer. This study did not address more in-depth data obfuscation like altering the registry, manually deleting the VM files and directories, or running a cleaner. This study stands as part of the initial discussion on the subject of virtualization forensics, not a definitive answer to the questions posed by the results of this research.

References

AccessData. (2013). Forensic Toolkit (FTK). (5.0.1). Retrieved April 2013, from AccessData: http://www.accessdata.com/support/product-downloads

AccessData. (2013, April 1). *Product Downloads* . Retrieved from AccessData: http://www.accessdata.com/support/product-downloads

AccessData Corp. (2007). Registry Viewer User Guide. Retrieved from http://marketing.accessdata.com/acton/attachment/4390/f-007b/0/-/-/-/-/file.pdf

Anglano, C. (2010). Forensic implications of virtualization technologies. *Handbook of Research on Computational Forensics, Digital Crime, and Investigation: Methods and Solutions*. (C. Li, Ed.) Retrieved from IGI Global: http://www.igi-global.com/chapter/forensic-implications-virtualization-technologies/39228

Anglano, C. (2012, May 30). *Cosimo anglano home page.* Retrieved from University of Eastern Piedmont Amedeo Avogadro: http://people.unipmn.it/mino/

Apriorit Inc. (2011, February 08). *Pros and Cons of Using Virtualization in Software Testing*. Retrieved from Apriorit: http://www.apriorit.com/qa-blog/223-virtualization-in-testing

Barrett, D. (2010, December 16). *Forensic challenges in virtualized environments.* Retrieved from University of Advanced Technology: http://www.uat.edu/academics/forensic_challenges_in_virtualized_environments.aspx

Bazargan, F., Yeun, C. Y., & Zemerly, M. J. (2012). State-of-the-art of virtualization, its security threats and deployment models. *International Journal for Information Security Research, Volume 2*(Issues 3/4). Retrieved from http://infonomics-society.org/IJISR/Paper 1.pdf

E-fense. (2013). *Helix 3 Pro*. Retrieved October 2013, from E-fense: http://www.e-fense.com/helix3pro.php

File-Extensions.org. (2013). *VirtualBox file extensions*. Retrieved from File-Extensions.org - The Source for File Extensions Information: http://www.file-extensions.org/virtualbox-file-extensions

Fitzpatrick, J. (2010, December 19). *Five best virtual machine applications*. Retrieved from LifeHacker: http://lifehacker.com/5714966/five-best-virtual-machine-applications

Garfinkel, S. (2007, March 09). *Anti-forensics: Techniques, detection and countermeasures.* Retrieved from Cite Seer X: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.109.5063&rep=rep1&type=pdf

Google Code. (2013). Regshot. (1.8.3). Retrieved October 2013, from http://code.google.com/p/Regshot/downloads/list

Hirwani, M., Pan, Y., Stackpole, B., & Johnson, D. (2012, July). Forensic acquisition and analysis of vmware virtual hard disks. *The 2012 international conference on security and management*. Las Vegas, NV. Retrieved from R.I.T Digital Media Library: https://ritdml.rit.edu/handle/1850/15922

Liston, T., & Skoudis, E. (2006). *On the cutting edge: Thwarting virtual machine detection.* Retrieved from SANS Internet Storm Center: http://handlers.sans.org/tliston/ThwartingVMDetection_Liston_Skoudis.pdf

Mehnert, F. (2012, October 11). [vbox-dev] aboutsav file created by snapshots [Online forum comment]. Retrieved from https://www.virtualbox.org/pipermail/vbox-dev/2012-October/010986.html

Messmer, E. (2009, October 20). *Gartner: Server virtualization now at 18% of server workload.* Retrieved from Network World: http://www.networkworld.com/news/2009/102009-gartner-server-virtualization.html

Moussat, G. (2012, August 21). *How to reuse virtualbox disk "snapshots" [Online forum comment]*. Retrieved from http://superuser.com/questions/464445/how-to-reuse-virtualbox-disk-snapshots

MPack. (2009, September 15). *CloneVDI-Discussion and Support [Online forum comment]*. Retrieved from virtualbox.org-End user forums for VirtualBox: https://forums.virtualbox.org/viewtopic.php?t=22422

NirSoft. (2013). FolderChangesView. (1.50). Retrieved October 2013, from http://www.nirsoft.net/utils/folder_changes_view.html

Oracle Corporation. (2013). Oracle vm virtualbox user manual. (4.2.16). Retrieved from http://download.virtualbox.org/virtualbox/UserManual.pdf

Oracle Corporation. (2013). *Oracle vm virtualbox user manual. (Version 4.2.16).* Retrieved from User manual: http://download.virtualbox.org/virtualbox/UserManual.pdf

Oracle Corporation. (2013). VirtualBox 4.2.18 for Windows hosts x86/amd64. Retrieved October 2013, from https://www.virtualbox.org/wiki/Downloads

Oracle Corporation. (2013). *Welcome to virtualbox.org!* Retrieved from VirtualBox.org: https://www.virtualbox.org/

Pavlov, I. (2013). 7-Zip Portable. Retrieved October 2013, from http://portableapps.com/apps/utilities/7-zip_portable

PortableApps.com. (2013). *Portableapps.com platform features*. Retrieved from PortableApps.com: http://portableapps.com/platform/features

Rogers, M. (2006, March 23). *Anti-Forensics: The Coming Wave in Digital Forensics.* Retrieved from The Center for Education and Research in Information Assurance and Security:

http://www.cerias.purdue.edu/news_and_events/events/symposium/2006/materials/pdfs/antiforensics.pdf

Shavers, B. (2008). *Virtual forensics: A discussion of virtual machines related to forensics.* Retrieved from Forensic Focus: http://www.forensicfocus.com/downloads/virtual-machines-forensics-analysis.pdf

Swanson, I., & Williams, P. A. (2008, December). Virtual environments support insider. *6th Australian digital forensics conference, Edith Cowan University, Perth Western Australia*. Retrieved from http://www.techrepublic.com/resource-library/whitepapers/virtual-environments-support-insider-security-violations/

Ubuntu. (2013). Ubuntu. (13.04 32-bit). Retrieved October 2013, from http://www.ubuntu.com/download/desktop

VirtualBoxes-Free VirtualBox ® Images. (2013). Ubuntu Linux. (12.10 x86). Retrieved October 2013, from http://virtualboxes.org/images/ubuntu/

VTask Studio. (2013). Whatchanged. (1.07). Retrieved October 2013, from http://www.vtaskstudio.com/support.php

Wallen, J. (2013, August 21). 10 ways to get the most from VirtualBox [Web log message]. Retrieved from http://www.techrepublic.com/blog/10-things/10-ways-to-get-the-most-from-virtualbox/

WinMD5. (2009). Winmd5free. Retrieved October 2013, from http://www.winmd5.com/