Investigating the Implications of Virtualization for Digital Forensics*

Zheng Song[1], Bo Jin[2], Yinghong Zhu[1], and Yongqing Sun[2]

[1] School of Software, Shanghai Jiao Tong University, Shanghai 200240, China
[2] Key Laboratory of Information Network Security, Ministry of Public Security, People's Republic of China (The Third Research Institute of Ministry of Public Security), Shanghai 201204, China
{songzheng,zhuyinghong}@sjtu.edu.cn, jinbo@stars.org.cn, yongqing.sun@gmail.com

Abstract. Research in virtualization technology has gained significant momentum in recent years, which brings not only opportunities to the forensic community, but challenges as well. In this paper, we discuss the potential roles of virtualization in the area of digital forensics and conduct an investigation on the recent progresses which utilize the virtualization techniques to support modern computer forensics. A brief overview of virtualization is presented and discussed. Further, a summary of positive and negative influences on digital forensics that are caused by virtualization technology is provided. Tools and techniques that are potential to be common practices in digital forensics are analyzed and some experience and lessons in our practice are shared. We conclude with our reflections and an outlook.

Keywords: Digital Forensics, Virtualization, Forensic Image Booting, Virtual Machine Introspection.

1 Introduction

As virtualization is becoming increasing mainstream, its usage becomes more commonplace. Virtual machines, so far, have a variety of applications. Governments and organizations can have their production systems virtualized to reduce costs on energy, cooling hardware procurements and human resources, enhance availability, robustness and utilization of their systems. Software development and testing is another field that virtual machines are widely used, because virtual machines can be installed, replicated and configured in a short time and support almost all existing operating systems, thus improving the productivity and efficiency. As for security researchers, a virtual machine is a controlled clean environment in which unknown codes from the wild are run and analyzed. Once an undo button is pressed, the virtual machine will roll back to the previous clean states.

* This paper is supported by the Special Basic Research, Ministry of Science and Technology of the People's Republic of China (No. 2008FY240200), and the Key Project Funding, Ministry of Public Security of the People's Republic of China (No. 2008ZDXMSS003).

X. Lai et al. (Eds.): E-Forensics 2010, LNICST 56, pp. 110–121, 2011.

While its benefits are attractive, virtualization also brings challenges to the digital forensics practitioners. With the advent of various virtualization solutions, a lot of work should be done to have a full understanding of all the techniques related with digital forensics. A virtual machine not only can be a suspect's tool for illegal activities, but also become a useful tool for forensic investigator/examiner. Recent years have witnessed a trend of virtualization as a focus in the IT industry and we believe it will have an irreversible influence on the forensic community and their practices as well.

In this paper, we analyze the potential roles that virtual machines will take and investigate several promising forensic techniques that utilize virtualization. A detailed discussion about benefits and limitations of these techniques is provided and lessons learned during our investigation are given.

The next section reviews the idea of virtualization. Section 3 discusses the scenarios where virtual machine is taken as suspect targets. Section 4 introduces several methods that regard virtual machines as forensic tools. We conclude with our reflections on this topic.

2 Overview of Virtualization

The concept of virtualization is not new but its resurgence came only in recent years. Virtualization provides an extra level of abstraction in contrast to the traditional architecture of computer systems, as illustrated in Figure1.

On a broader view, virtualization can be categorized into several types including ISA level, Hardware Abstraction Layer (HAL) level, OS level, Programming language level and Library level, according to the different layer in the architecture where virtualization layer is inserted. HAL-level virtualization, also known as system level virtualization or hardware virtualization, allows the sharing of underlying physical resources between different virtual machines which are based on the same ISA (e.g., x86). Each of the virtual machines is isolated between others and runs its own operating system.

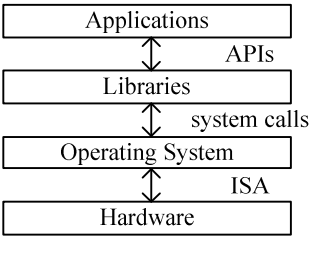

Fig. 1. The hierarchical architecture of modern computer systems

The software layer that provides the virtualization abstraction is called virtual machine monitor (VMM) or hypervisor. Based on the diverse positions where it is implemented, VMM, or hypervisor, can be divided into Type I, which runs on bare metal and Type II, which runs on top of an operating system.

In a Type I system, the VMM runs directly on physical hardware and eliminates an abstraction layer (i.e., host OS layer), so the performance of Type I virtual machines overwhelms that of Type II in general. But Type II systems have closer ties with the underlying host OS and their device drivers; they often have a wider range of functionalities in physical hardware components. This paper involves mainstream virtualization solutions, such as VMware Workstation [39], VMware ESXi [38], and Xen [29]. Figure 2 shows those two architectures. Xen and VMware ESXi belong to the former and VMware Workstation the latter.

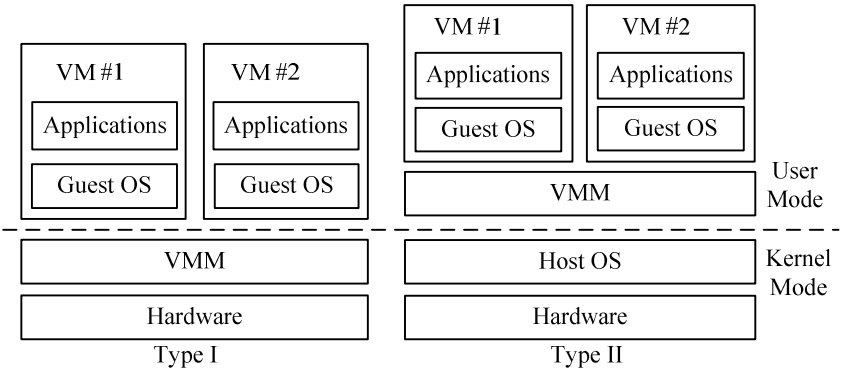

Fig. 2. Different architectures of VMMs, *Type I* on the left and *Type II* on the right

3 Virtual Machines as Suspect Targets

A coin has two sides. With the wide use of virtual machines, it becomes inevitable that virtual machines may become suspect targets for forensic practitioners. The following will present the challenges and problems faced with the forensic society that are found during our research.

3.1 Looking for the Traces of Virtual Machines

The conventional computer forensics process comprises a number of steps, and it can be broadly encapsulated in four key phases [25]: access, acquire, analyze and report. The first step is to find traces of evidences.

There are a variety of virtualization solution products available, not only commercial, but open source and freeware as well. Many of these products are required to be installed on a host machine (i.e., Type II). For these types of solutions, in most cases, it is the simplest situation that both the virtual machine application and virtual machines existing on the target can be found directly. But occasionally, looking for the traces of virtual machines may become a difficult task.

Considering some deleted virtual machines or uninstalled virtual machine applications, they are attractive to examiners, although they are not typically considered as suspicious. Discovering the traces involves careful examination of remnants on a host

system: .lnk files, prefetch files, MRU references, registry and sometimes special files left on the hard drive. Shavers [17] showed some experience in looking for the traces: the registry will most always contain remnants of program install/uninstall as well as other associated data referring to virtual machine applications; file associations maintained in the registry will indicate which program will be started based upon a specific file being selected; the existence of "VMware Network Adaptor" without the presence of its application can be a strong indication that the application did exist on the computer in the past. In the book [23], Chapter 5 analyzed the impact of a virtual machine on a host machine. Virtual machines may be deleted directly by the operating system due to its size in Windows, and with today's data recovery means, it might be possible to recover some of these files, but impossible to examine the whole as a physical system. In a nutshell, this kind of recovery work is filled with uncertainty and the larger the size of the virtual machine is, the harder it is to recover in our experiments.

However, with other types of virtualization solutions (Type I), it is totally different to search for traces. For instance, as the Virtual Desktop Infrastructure (VDI) develops, desktop virtualization will gain more popularity. Virtual machine instances can be created, snapshot and deleted quickly and easily, and also can dynamically traverse through the network to different geographical locations. It is similar to the cloud computing environment where you hardly know on which hard disk your virtual machine resides in. Of the above circumstances, maybe only the virtualization application itself knows the answer. Even if you may find a suspect target through tough and arduous work, it could be of a previous version and contains no evidences you want at all. So searching for the existence of the very target is a prerequisite before further investigation is conducted, and it is a valuable field for forensic researchers and practitioners.

It is also important to notice that some virtualization applications do not need to be installed in a host computer and can be accessed and run in external media, including USB flash drivers or even CDs. It is typically considered as an anti-forensic method if he or she wants to disrupt the examinations.

3.2 Acquiring the Evidence

The acquisition of evidence must be conducted under a proper and faultless process; otherwise it will be questionable in court. The traditional forensic procedure, known as static analysis, is to take custody of the target system, shut it down, copy the storage media, and then analyze the image copy using a variety of forensics tools. The shutdown process amounts to either invoking the normal system shutdown sequence, or pulling the power cord from the system to effect an instant shutdown [19].

Type II virtual machines are easier to image, as they typically reside in one hard disk. In theory and practice, there may be more virtual machines in a single disk and a virtual machine may have close ties with the underlying host operating system, such as shared folders and virtual networks. Imaging the “virtual disk” only may miss evidences of vital importance in the host system. It is recommended to image the whole host disk for safety if possible, rather than image the virtual disk only.

An alternate way is to mount the VMDK files of VMware as mounted drives through VMware DiskMount Tool [16], instead of imaging the whole host system. In this way,

we can have access to these virtual disks without any VMware applications installed. Being treated as a drive, the virtual disk files can be analyzed with suitable forensic tools. However, it is better to mount a VMDK virtual disk on a write protected external media, which is recommended by Brett Shavers [17]. And further, we believe it is better to use this method if and only if all the evidences exists just in the guest OS, and this situation may be infrequently met.

However, for the Type I virtual machines which are commonly stored in large storage media such as SAN and NAS in production systems in enterprises, the traditional forensic procedure is improper and inappropriate now, as under these circumstances, it is neither practical nor flawless to acquire the evidence in an old fashion: powering off the server could lead to unavailability to other legal users thus become involved in several issues.

The most significant one is the legislative issues as who on earth will account for total losses for the innocents. But we will not continue with it as it is not the focus of this paper. Besides, there are technical issues as well. For example, Virtual Machine File System (VMFS) [20] is a proprietary file system format owned by VMware, and there is a lack of forensic tools to parse this format thoroughly, which brings difficulties for forensic practitioners. What is worse, VMFS is a clustered advanced file system that a single VMFS file system can spread over multiple servers. Although there are some efforts in this field like open source VMFS driver [21], which enables read-only access to files and folders on partitions with VMFS, it is far from satisfying forensic needs. Even if the virtual machine can be exported to an external storage media, it may still arouse suspicions in court as it is reliant on cooperation from the VM administrator and also the help of virtualization management tools. In addition, as we have mentioned earlier, an obstacle to acquire the image of a virtual machine may be in the cloud-computing-alike situation where its virtual disk locates on different disks and has a huge size that imaging it with current technology faces more difficulty.

We also want to point out here that acquiring the virtual machine related evidence with traditional forensic procedure might not be enough or even might be questionable. In the case of a normal shut down of a VM, data is read and written to the virtual hard disk, which may delete or overwrite forensically relevant contents (similar things happens when shut down a physical machine). Another more important aspect lies in that much of the information, such as process list, network ports, encryption keys, or some other sensitive data, may only exist in RAM and it will not appear in the image.

It is recommended to perform a live forensic analysis on the target system in order to get particular information, the same with virtual environments. But note that live forensic analysis virtually faces its own problems and it is discussed in the next section.

3.3 Examining the Virtual Machine

The examination of a virtual machine image is almost the same with that of physical machine, with little differences. The forensic tools and processes are alike. The examination of a virtual machine incurs additional analysis of its related virtual machine files in the perspective of the host OS. The metadata associated with these file may give some useful information.

If further investigation on the associated virtual machine files continues, more detail about the moment when the virtual machine is suspended or closed may be revealed. Figure 3 shows the details of a .vmem file, which is a backup of the virtual machine's paging file. In fact, we believe it is a file storing the contents of "physical" memory. As we know, the virtual addresses used by programs and operating system components are not identical with the true locations of data in physical memory image (dump). It is the examiner's ability to translate the addresses [24]. In our view, the same technique applies to the memory analysis of virtual machines.

It is currently a trend to perform a live forensics [22] when a computer system to examine is in a live state. Useful information of the live system at the moment, such as memory contents, network activities and active process lists will probably not survive after the system is shut down. It is possible to encounter that a live system to be examined involves one or more running virtual machines as well. Running processes or memory contents of a virtual system may as important as, or even more important than that of the host system. But it is highly likely that performing live forensic in the virtual machine will almost certainly affect not only the states of the guest system but also the host system. There is less experience in this situation from literature and we believe it must be tackled carefully.

In addition, encryption is a traditional barrier in front of forensic experts during examination. In order to protect privacy, more and more virtualization providers tend to introduce encryption, which consequently arise the difficulties. This is a new trend which more attentions should be paid to.

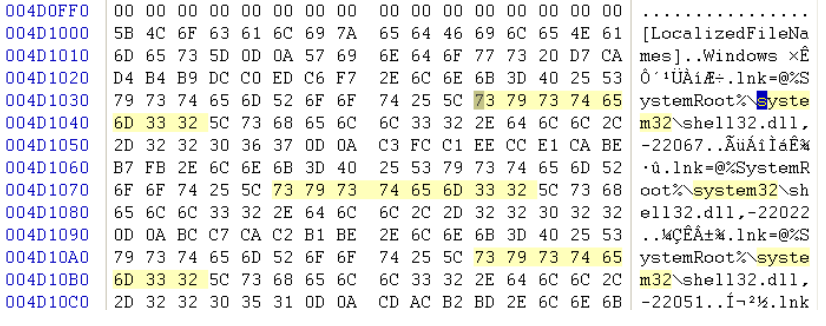

Fig. 3. The contents of a .vmem file which may include some useful information. A search for the keyword "system32" returned over 1000 hits in a .vmem file of Windows XP virtual machine, and the above figure just show some of them as an example.

4 Virtual Machines as Forensic Tools

Virtualization provides new technologies that promote our forensic tool boxes and we now have more methods in proceeding with the examination. We have focused our attention on the following two fields, forensic image booting and virtual machine introspection.

4.1 Forensic Image Booting

Before forensic image booting with virtual machine comes up, restoration of a forensic image back to disk requires numerous attempts, if the original hardware is not available. And blue screens of death are frequently met. However, with virtual machines solutions, our burden relieves. A forensic image can be booted in a virtual environment, with less manual work as clicking the mouse and the left work is done automatically.

The benefits of booting up a forensic image are various. The obvious one is that it benefits forensic examiners by quick and intuitive insight into the target, which can save a lot of time if nothing valuable exists. Also it provides examiners a convenient way to demonstrate the evidence to the non-experts in the court in a view that is as if seen by the suspect by the time to seizure.

Booting a forensic image requires certain steps. Depending on the format of the image, different tools are prepared. Live View [1] is a forensics tool produced by CERT that creates a VMware virtual machine out of a raw disk image (dd-style) or physical disk. In our practice, dd format and Encase EWF format are mostly used. Encase EWF format (E01) is a proprietary format that is commonly used worldwide and includes additional metadata such as case number, investigator's name, time, notes, checksum and footprint (hash values). Besides, it can reside in multiple segment files or within a single file. So it is not identical with the original hard disk and can not be boot up directly. To facilitate the booting, we developed a small tool to convert Encase EWF files to dd image. Figure 4 illustrates the main steps we use in practice.

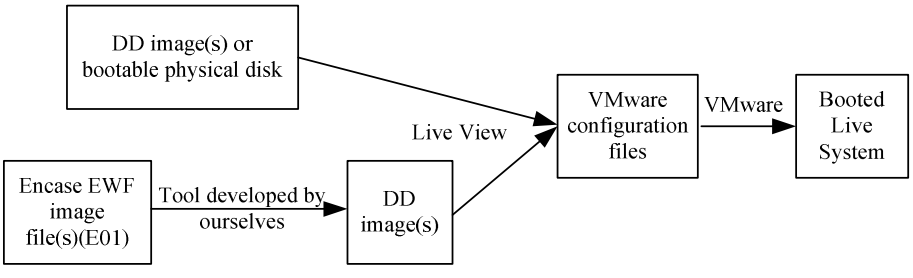

Fig. 4. The main steps to boot forensic image(s) up in our practice

It is recommended to use write-protected devices for safety, in case there would be unexpected accidents. With the support from Live View, investigators can interact with the OS inside the forensic image or physical disk without modifying the evidence, because all the changes to the OS is written to separate virtual machine files, not the original place. Repeated and separate investigations are now available.

Other software tools that can create the files with parameters for virtual machine include ProDiscover Basic [11] and Virtual Forensics Computing [12]. An alternate

method to deal with the forensic images with proprietary format is to mount these forensic images as disks beforehand using tools such as Mount Image Pro [13], Encase Forensics Physical Disk Emulator [14] and SmartMount [15].

Based on this forensic image booting technique, a lot of work is done. Bem et al. [10] proposed a new approach where two environments, conventional and virtual, are used independently. After the images are collected in a forensically sound way, two copies are produced. One is protected using the chain of custody rules, and the other is given to a technical worker who works with it in virtual environments. Any findings are documented and passed to a more qualified person who confirms them in accordance with forensic rules. They demonstrated that their approach can considerably shorten the time of the computer forensic investigation analysis phase and allow for better utilization of less qualified personnel.

Mrdovic et al. [26] proposed combinations of static and live analysis. Virtualization is used to bring static data to life. Using data from memory dump, virtual machine created from static data can be adjusted to provide better picture of the live system at the time when the dump was made. Investigator can have interactive session with virtual machine without violating evidence integrity. And their tests with sample system confirm viability of their approach.

As a lot of related work [10, 26, 27] shows, forensic image booting seems to be a promising technology. However, we have found that there exist some anti-forensic methods in the wild during our investigation. One of them is to utilize a small program which uses the virtual machine detection code [2] to shut the system down as soon as a virtualized environment is detected during system startup. Although investigators may finally figure out what has happened and remove this small program to successfully boot the image, extra efforts are made and more time wasted. But this raises our concerns about the covert channels in virtualization solutions, which is still a difficult problem to deal with.

4.2 Virtual Machine Introspection

As we have mentioned before, live analysis has particular strengths over traditional static analysis. But still, live analysis has its own limitations. One limitation, as we have discussed in Section 3.2, which is also known as the observer effect, is that any operation performed during the live analysis process modifies the state of the system, which in turn might result in potential contamination to evidences. The other limitation, as Brian D. Carrier analyzed, is that the current risks in live acquisition [3] lie in the systems to be examined are themselves compromised or incomplete (e.g., by rootkits). Further more, any forensic utilities executed during the live analysis can be detected by a sufficiently careful and skilled attacker, who can at that point change behavior, delete important data, or actively obstruct the investigator's efforts [28]. In that case, live forensic may output inaccurate or even false information. Resolving these issues depends on forensic experts themselves. However, using virtual machines and the Virtual Machine Introspection (VMI) technique, the above limitations may be overcome.

Suppose a computer system runs in a virtual machine, which is supervised by a virtual machine monitor. As VMM has complete read and write access to all memory in VM (in most cases), it is possible for a special tool to reconstruct the contents of a process's memory space, and even the contents of the VM's kernel memory, by using the page table for the VMM and its privileges to obtain an image of the VM's memory. This special tool will gain all memory contents of interest, thus help to fully understand what the target process was doing for the purpose of forensic analysis. The above is just an illustration of the usage of virtual machine introspection and more functionality are possible such as monitoring disk accesses and network activities.

One of the nine research areas identified in the virtualization and digital forensics research agenda [4] is virtual introspection. Specifically, Virtual Machine Introspection is the process by which the state of a virtual machine is observed from either the Virtual Machine Monitor, or from a virtual machine other than the one being examined. This technique was first introduced by Garfinkel and Rosenblum [5].

Research in the application of VMI has typically focused on intrusion detection rather than digital forensics [6]. But there are some associated work in the forensic filed recently. XenAccess [7] project, led by Bryan Payne from Georgia Tech, produced an open source virtual machine introspection library in Xen hypervisor. This library allows a privileged domain to view the runtime state of another domain. It currently focuses on memory access, but also provides proof-of-concept code for disk monitoring. Brian Hay and Kara Nance [8] provide a suite of virtual introspection tools for Xen (VIX tools), which allow an investigator to perform live analysis of an unprivileged Xen [29] virtual machine (DomU) from the privileged Dom0 virtual machine. VMwatcher [30], VMwall [31], and others [32, 33] were developed to monitor VM execution and infer guest states or events, and all of them provide the potential ability to be used in forensics.

However, it seems there is a lack of similar tools in the bare-metal architecture (Type I) solutions of commercial products. Most recently, VMware has introduced VMsafe [9] technology that can allow third-party security vendors to leverage the unique benefits of VMI to better monitor, protect and control guest VMs. But VMsafe mainly addresses security issues, not forensic ones. We believe that VMsafe technology, if gained cooperation with VMware, could be changed and ported to a valuable forensic tool suite on VMware platform.

Nance et al. [28] identified four initial priority research areas in VMI and discussed its potential role in forensics. Virtual Machine Introspection may help the digital forensics community, but it still needs time to be proved and applied, as digital forensics investigation must be serious. We are cautious as we believe that time tries all things. Luckily, our cautions are proved right!

Bahram et al. [18] implemented a proof-of-concept Direct Kernel Structure Manipulation (DKSM) prototype to subvert the VMI tools (e.g., XenAccess). The exploit relies on the assumption that the original kernel data structures are respected by the distrusted guest and thus can directly used to bridge the well-known semantic gap [34]. The semantic gap can be explained as follows: from outside the VM, we can get a view of the VM at the VMM level, which includes its register values, memory pages, disk blocks; whereas from inside the VM, we can observe semantic-level entities

(e.g., process and files) and events (e.g., system calls). This semantic gap is formed by the vast difference between external and internal observations. To bridge this gap, a set of data structures (e.g., those for process and file system management) can be used as "templates" to interpret VMM-level VM observations.

We believe current Virtual Machine Introspection has at least several limitations:

The first one is its trustiness. A VMI tool aims to analyze a VM which is not trusted, but still expects a VM to respect the kernel data structure templates, and relies on the VM maintained memory contents. Fundamentally, this is a trust inversion in logic. For the same reason, Bahram et al. [18] believe existing memory snapshop-based memory analysis tools and forensics systems [35, 36, 37] share the same limitation.

The second one is its detectability. There are several possibilities: (1) Timing analysis, as analysis of a running VM typically requires a period of time and might cause an inconsistent view. So a pause to a running VM might be unavoidable, thus might be detectable; (2) Page faults analysis [8], as the VM may be able to detect unusual patterns in the distribution of page faults, caused by the VMI application accessing pages that have been swapped out, or causing pages that were previously swapped out to be swapped back into RAM.

So moving toward the development of next-generation, reliable Virtual Machine Introspection technology is the future direction for researchers interested in this field.

5 Conclusion

On the wave of virtualization, forensic community should adapt themselves to new situations. On one hand, as we have discussed earlier, criminals may use virtual machines as handy tools and desktop computers might be replaced with thin clients in enterprise in the near future; all these will undoubtedly add the difficulties in the forensic process and we should prepare for them. On the other hand, virtualization provides us with new technique that can facilitate the forensic investigation, such as the forensic image booting. However, these techniques should be introduced into this domain carefully with overall tests, as digital forensics can have serious and significant legal and societal consequences.

This paper describes several forensic issues that come along with virtualization and virtual machines, provides experience and lessons in our research and practice.

References

1. Live View, `http://liveview.sourceforge.net/`
2. Detect if your program is running inside a Virtual Machine, `http://www.codeproject.com`
3. Carrier, B.D.: Risks of Live Digital Forensic Analysis. Communications of the ACM 49, 56–61 (2006)
4. Pollitt, M., Nance, K., Hay, B., Dodge, R., Craiger, P., Burke, P., Marberry, C., Brubaker, B.: Virtualization and Digital Forensics: A Research and Education Agenda. Journal of Digital Forensic Practice 2, 62–73 (2008)

5. Garfinkel, T., Rosenblum, M.: A virtual machine introspection based architecture for intrusion detection. In: 10th Annual Symposium on Network and Distributed System Security, pp. 191–206 (2003)
6. Nance, K., Bishop, M., Hay, B.: Virtual Machine Introspection: Observation or Interference? IEEE Security & Privacy 6, 32–37 (2008)
7. XenAccess, http://code.google.com/p/xenaccess/
8. Hay, B., Nance, K.: Forensic Examination of Volatile System Data using Virtual Introspection. ACM SIGOPS Operating Systems Review 42, 74–82 (2008)
9. VMsafe, http://www.vmware.com
10. Bem, D., Huebner, E.: Computer Forensic Analysis in a Virtual Environment. International Journel of Digital Evidence 6 (2007)
11. ProDiscover Basic, http://www.techpathways.com/
12. Virtual Forensics Computing, http://www.mountimage.com/
13. Mount Image Pro, http://www.mountimage.com/
14. Encase Forensics Physical Disk Emulator, http://www.encaseenterprise.com/
15. SmartMount, http://www.asrdata.com/SmartMount/
16. VMware DiskMount, http://www.vmware.com
17. Shavers, B.: Virtual Forensics (A Discussion of Virtual Machine Related to Forensic Analysis), http://www.forensicfocus.com/virtual-machines-forensics-analysis
18. Bahram, S., Jiang, X., Wang, Z., Grace, M., Li, J., Xu, D.: DKSM:Subverting Virtual Machine Introspection for Fun and Profit. Technical report, North Carolina State University (2010)
19. Carrier, B.: File system forensic analysis. Addison-Wesley, Boston (2005)
20. VMFS, http://www.vmware.com/products/vmfs/
21. Open Source VMFS Driver, http://code.google.com/p/vmfs/
22. Farmer, D., Venema, W.: Forensic Discovery. Addison-Wesley, Reading (2005)
23. Dorn, G., Marberry, C., Conrad, S., Craiger, P.: Advances in Digital Forensics V. IFIP Advances in Information and Communication Technology, vol. 306, p. 69. Springer, Heidelberg (2009)
24. Kornblum, J.D.: Using every part of the buffalo in Windows memory analysis. Digital Investigation 4, 24–29 (2007)
25. Kruse II, W.G., Heiser, J.G.: Computer Forensics: Incident Response Essentials, 1st edn. Addison Wesley Professional, Reading (2002)
26. Mrdovic, S., Huseinovic, A., Zajko, E.: Combining Static and Live Digital Forensic Analysis in Virtual Environment. In: 22nd International Symposium on Information, Communication and Automation Technologies (2009)
27. Penhallurick, M.A.: Methodologies for the use of VMware to boot cloned/mounted subject hard disk image. Digital Investigation 2, 209–222 (2005)
28. Nance, K., Hay, B., Bishop, M.: Investigating the Implications of Virtual Machine Introspection for Digital Forensics. In: International Conference on Availability, Reliability and Security, pp. 1024–1029 (2009)
29. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T.L., Ho, A., Neugebaur, R., Pratt, I., Warfield, A.: Xen and the art of virtualization. In: Nineteenth ACM Symposium on Operating Systems Principles, pp. 164–177. ACM Press, New York (2003)
30. Jiang, X., Wang, X., Xu, D.: Stealthy malware detection through vmm-based "out-of-the-box" semantic view reconstruction. In: 14th ACM conference on Computer and communications security, Alexandria, Virginia, USA, pp. 128–138 (2007)

31. Srivastava, A., Giffin, J.: Tamper-resistant, application-aware blocking of malicious network connections. In: 11th International Symposium on Recent Advances in Intrusion Detection, pp. 39–58. Springer, Heidelburg (2008)
32. Jones, S.T., Arpaci-Dusseau, A.C., Arpaci-Dusseau, R.H.: Antfarm: tracking processes in a virtual machine environment. In: Annual Conference on USENIX 2006 Annual Technical Conference, p. 1. USENIX Association, Berkeley (2006)
33. Litty, L., Lagar-Cavilla, H.A., Lie, D.: Hypervisor support for identifying covertly executing binaries. In: 17th Conference on Security Symposium. USENIX Association (2008)
34. Chen, P.M., Noble, B.D.: When virtual is better than real. In: Eighth Workshop on Hot Topics in Operating Systems, p. 133. IEEE Computer Society, Washington, DC (2001)
35. Volatile systems, https://www.volatilesystems.com/default/volatility
36. Carbone, M., Cui, W., Lu, L., Lee, W., Peinado, M., Jiang, X.: Mapping kernel objects to enable systematic integrity checking. In: 16th ACM Conference on Computer and Communications Security, pp. 555–565. ACM, New York (2009)
37. Dolan-Gavitt, B., Srivastava, A., Trayor, P., Giffin, J.: Robust signatures for kernel data structures. In: 16th ACM Conference on Computer and Communications Security, pp. 566–577 (2009)
38. VMware ESXi, `http://www.vmware.com/products/esxi/`
39. VMware Workstation, `http://www.vmware.com/products/workstation/`