2008

Virtual Environments Support Insider Security Violations

Iain Swanson
Edith Cowan University

Patricia A.H. Williams
Edith Cowan University

DOI: 10.4225/75/57b27c4940cc4
Originally published in the Proceedings of the 6th Australian Digital Forensics Conference, Edith Cowan University, Perth Western Australia, December 3rd 2008.

Virtual Environments Support Insider Security Violations

I. Swanson and P.A.H. Williams

SECAU Security Research Centre
Edith Cowan University

Abstract

This paper describes an investigation into how an employee using a virtual environment can circumvent any or all of the security, policies and procedures within an organization. The paper discusses the fundamental issues that organizations must address to be able to detect such an attack. Attacks of this nature may be malicious with intent to cause disruption by flooding the network or disabling specific equipment, or non-malicious by quietly gathering critical information such as user names and passwords or a colleague's internet banking details. Identification of potential residual evidence following an attack is presented. Such evidence may be used to speculate or verify an attack incident occurrence. Additionally, the forensic extraction of any such evidence is discussed. Finally, the paper raises the possibility of a virtual machine being used as an anti-forensic tool to destroy incriminating evidence in such circumstances.

Keywords: Anti-Forensics, computer forensics, virtual machine, security, antivirus

INTRODUCTION

The security and protection of an organizations data and infrastructure is of utmost importance to all companies regardless of size. Tens of thousands and in some cases millions of dollars are being spent each year to try to keep networks and computers free of infection and secure from outside intruders. There are numerous threats from individuals trying to compromise an organization's security from beyond their exterior firewalls, however the "attacker" working from within the organization should also be considered. Is there enough security inside to protect the organization? This paper will look at a real life situation to try and find the answers to these questions within the bounds of legality. In general, the desktop standard operating environment (SOE) is built to include basic security tools such as 'antivirus', 'sms client', 'vnc' and a locked down 'desktop firewall'. Firewalls are placed between different environments, Secure Lan, DMZ's and so on. Intrusion detection systems on servers, network monitoring, security on switches and routers, port lock downs, web content filtering and mail checking should provide adequate protection for any organization. The main problem with security measures such as these, are that they can frustrate employees if they cannot get to their favourite web sites, or their personal mail gets bounced. This in turn may turn an employee into a security risk as they search for loop hole in the system. The other scenario however and the most likely one, is that the organization already has an attacker in their environment looking to circumvent security by using a virtual machine to get round the policies and security measurers that are in place. "These intra-host threats can elude any existing security protection schemes" (Ruykhaver, 2002).

Assumptions

Several assumptions are made about the type of access to the computer and the skill level the employee may have. It is assumed that the employee has or can obtain administrator access to the machine that will be used for any attacks. It is also thought that the employee would have sufficient technical knowledge to install software and possesses the skill to use it. In this case the potential attack under investigation is in deference to the employee being able to bring in a laptop into the business environment and connecting it directly into the network, as this may be noticed.

CIRCUMVENTION

In recent years, the proliferation of virtual environments has been accelerated due to products like VMware Server becoming available to download at no cost (VMware Inc, 2008). Other advancements, such as VMplayer and pre-built 'virtual appliances', have made it straightforward to install and run a separate operating aystem with a complete set of exploit, monitoring, sniffing and cracking tools within the normal desktop environment.

Virtual machine

Virtualization provides a layer of abstraction between computing, storage and networking hardware of the host machine and the applications that run on it. This provides the virtual machine with an operating environment identical to the host on which it sits without knowing anything about the host. The virtual machine is completely isolated from the host although it shares the same hardware. The host machine only knows that an application is running and it is similarly unaware of another operating system. Figure 1 shows a before and after virtualization overview (VMware Inc, 2008).

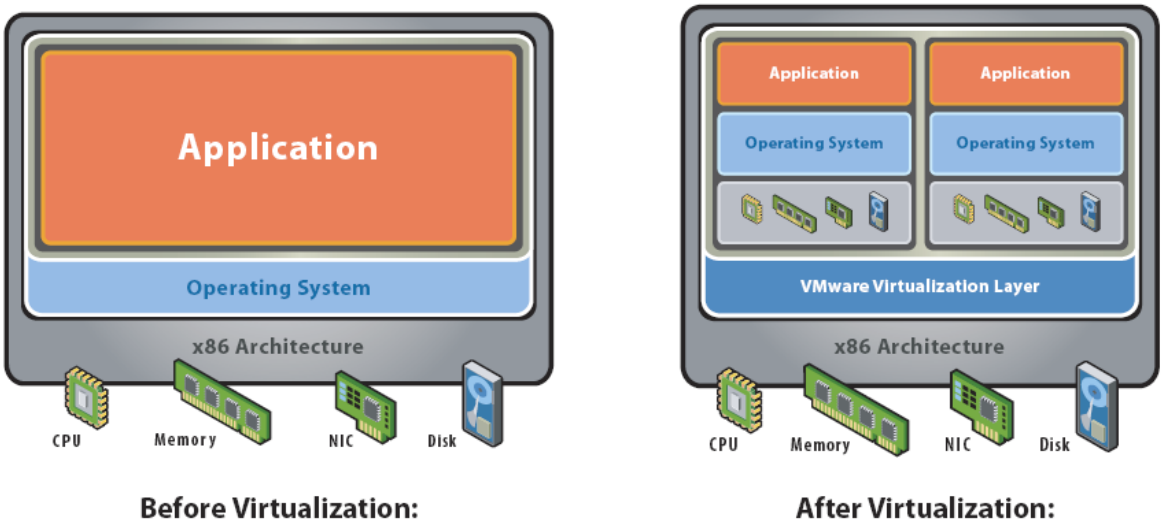

Figure 1. A virtualisation overview (VMware Inc, 2008.)

Using VMware Player and a pre-made virtual machine such as Backtrack3 (Remote Exploit, 2008), a new operating system can be booted up. The network interface card is set to bridge network address translation giving assess to explore the network with tools that are now invisible to the antivirus engine on the desktop. In normal use, tools like 'John the Ripper' (Openwall Project, n.d.) or 'dniff'(Song, 1999) would be deleted or at least quarantined by the antivirus software. Further, if the system is monitored it would raise an alarm that suspect tools were in use. Using a pre-made Linux distribution allows the use of the environment without additional complication. This creates a hole in security for the organization and any tool can be downloaded to the virtual machine undetected unless blocked by a web filter. Subsequently, this block can also be circumvented by using HTTP tunnelling or using an open SSH. In the example described below SSH is blocked so no SSH port redirects can be done. This using PingFu Iris (PingFu, 2008) a secure tunnel to the Internet is created.

Attacks

The infrastructure within the organization is now at risk from this virtual machine not only from the attacker but from the covert attacker that may now have access on the back of other software downloaded from the Internet. The host machine itself is safe from viruses, trojans and other malware, and the virtual machine is encapsulated and cannot infect the host directly, but the virtual machine can infect other devices on the network.

Backtrack3 is a comprehensive suite of the tools necessary to exploit any device not secured on the network. Attacks can be launched at specific machines or devices. Also, 'nmap' can be deployed to discover computers on the network and to see if it is detected by the security teams (Lyon, 2008).

One aim of such attacks is to listen for conversations between users and other devices or websites that use clear text user names and passwords. One method of doing this is to use Ettercap in promiscuous mode and ARP poisoning (Ornaghi & Valleri, 2005). Ettercap is a very powerful tool that is able to sniff switched networks for passwords and even redirect web sessions to different web sites. Such abilities could be used to fake internet banking sites for instance. The ARP cache poisoning attack works by poisoning the ARP cache of the target hosts (Spangler, 2003). Figure 2 below shows how ARP spoofing works.

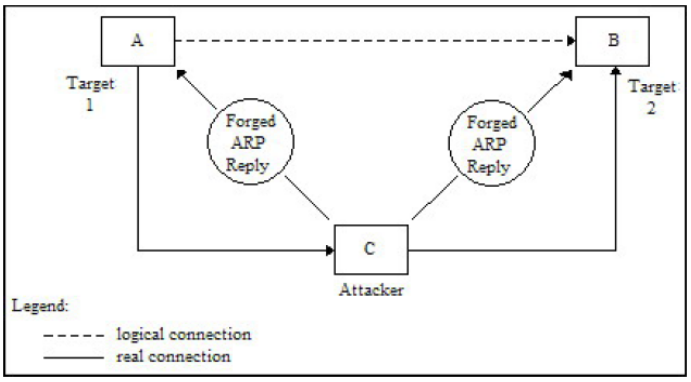

Figure 2. ARP spoofing (Spangler,2003)

Backtrack 3 has a comprehensive list of tools that can be used to exploit a computer or network. Included in the suite of tools is Metasploit Framework which has many types of exploits and pay loads that can be implemented with relative ease. Metasploit Framework is described as "a development platform for creating security tools and exploits" (Metasploit, 2008).

Trialling

Using the methods described above, a trial of the potential attacks was undertaken. By using a combination of tools built into the virtual machine's operating system it was discovered that user names and internet passwords, telnet user names, and password for routers and switches when 'SSH' was not being used, could be recovered. Potentially, malicious attacks could be run from this platform without immediate detection. HTTP sessions could be high-jacked, attacks on routers and switches, and compromises to servers holding sensitive data that the employee does not have legitimate access to, could be effected. As this was a live test and to ensure that no undue security risk was put in place the testing stopped. Whilst not definitive, it was enough to prove that an organization can be compromised by an installation of a virtual environment.

PREVENTION AND DETECTION

Pre Attack Detection and Prevention

Consideration needs to be given as to whether or not such an installation and the use of tools can be detected and therefore prevented or at least intercepted. Several scenarios for detection are possible:

- If a virtual machine were in bridged mode the DHCP server would supply a new IP address. Therefore an administrator checking the logs would see a machine name with no domain. Whilst not significant, this should warrant further investigation, as it would show that it had the same MAC address as another machine currently on the network;
- Similarly, this scenario would also hold true if the network switches were monitored. The switch port would have one MAC address with two IP addresses bound to it. As Microsoft's SMS is installed as part of the SOE, regular scans can be made of all machines which would produce a list of all software on the desktop. This could then be followed up with a visit to the offending computer;
- Network monitoring can detect the presence of an ARP poisoner or spoofer if the monitors are configured to search for such software. Also, most modern switches can be configured to avoid using ARP or to statically map IP addresses to MAC's. In addition, there are many ways to secure the switch which outside the scope of this paper.
- ArpWatch is a utility that keeps a list of IP address to MAC address mappings and will contact the administrator if any of these mappings are altered. Any malicious attacks should be picked up by the internal IDS. In this scenario, IDS is only installed on critical servers so other data servers are vulnerable.
- Service packs should be up-to-date and a policy of renewal should be in place;
- From a technical perspective, machines could be locked down so that no additional software can be loaded, although this may cause a problem of usability and raises other security issues. Also the netlogon script could contain a script that would track and report an installation on the machine; and

- Usage monitoring could identify a problem if a machine seems to be producing large amount of data.

Discovery and Identification

One detection problem is that if the attacker is not creating significant 'noise' on the network or creating any suspicion, then any activity may largely go unnoticed. Thus it is likely that an attacker may continue to gather information until discovered by accident. There are several possible reasons for lack of detection: either the security team do not have the time to investigate minor anomalies in infrastructure, or as in the scenario discussed here, the internal infrastructure is large and the management diverse, such that problems remain unnoticed for some time.

Therefore, it is worth considering that if the attacks were noticed, or that ARP spoofing was being monitored, the attacker would be discovered. Essentially this would all depend on the quality of the documentation and how well patch panels are labelled. The attacker could be traced back to the switch that originated the attacks. Consequently, the port being used could be found and traced back to the panel, and eventually back to the RJ45 plug that is attached to the wall. Obviously, there are methods to circumvent this such as running a virtual machine from a machine which belongs to an absent employee and using a virtual network to access it. Alternatively, an attacker can spoof a MAC address of a machine that is switched off, it would slow detection, perhaps long enough to try and destroy any tracks or evidence on the host computer.

Investigation

After discovery and eventual identification of the computer involved in the attacks, a problem then exists to find evidence that will prove that this machine and its user were the culprit. The computer would need to be seized for analysis.

FORENSIC EVIDENCE

Forensic evidence must be obtained in a sound manner that can be verified and must be significant enough to provide a 'no reasonable doubt' in the minds of jurors or in this case a manager's mind. In this scenario, through the trialling of the potential attack method, evidence was able to be collected as described in the subsequent sections.

Evidence

A forensically sound image was taken of the suspected attackers computer disk using a live Linux distribution. This was then mounted and searched for any suspect tools. None were found. From the image there were traces of VMware Player on the disk i.e. the empty folder was still there. Figures 3 and 4 show what was found to be on the disk.

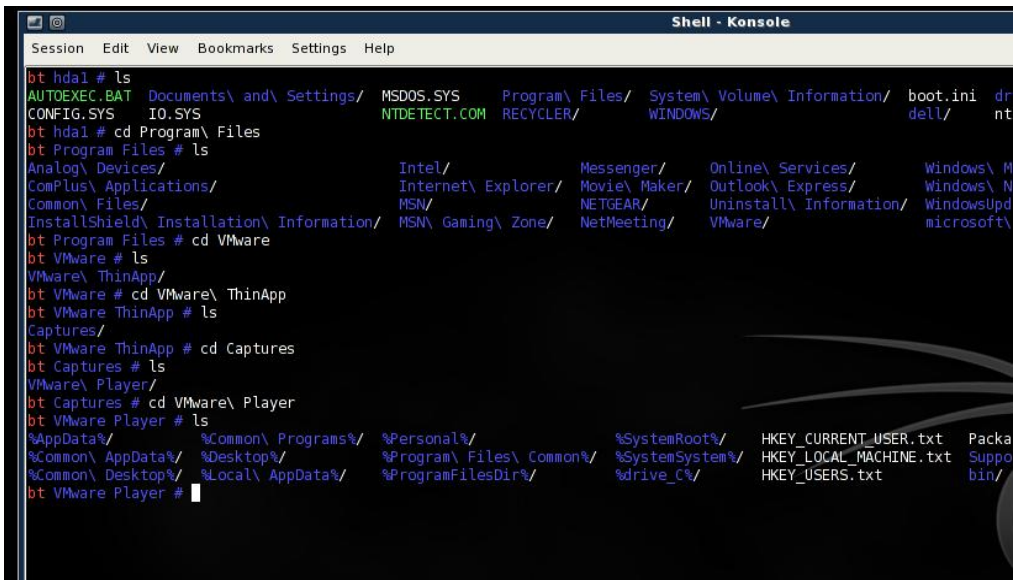

Figure 3. Residual evidence of VMware player installation.

Figure 3 shows that although VMplayer had been uninstalled, it was previously installed or at least some thing with the same name is present. The other directory 'ThinApp' contains a capture directory which turns out to be a new version of the Player called ThinApp.

The next step was the investigation of the rest of the disk and in particular looking at the Recycle bin to see if anything has been left there. Figure 4 shows its content.

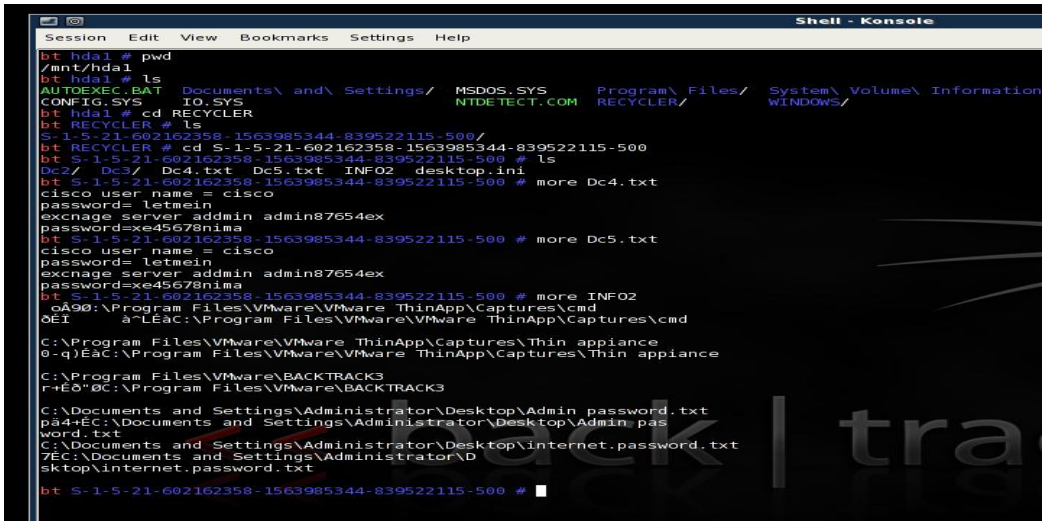

Figure 4. Recycle bin content.

Figure 4 shows there are several files left in the Recycle bin. Using the command 'more' it is possible to read the text files contained in it. The file INFO2 contains what appears to be a deleted copy of Backtrack. Backtrack 3 was extractable from the Recycle bin and contained the files necessary to run up a virtual machine. This evidence provides the ability to investigate the virtual disk.

This scenario, although done on a test bed, outlines some of the possibilities. In the example given the size of the virtual machine is only 3.5GB. If the image were much larger there may be problems of how to retrieve it. In a virtual machine there are several files that would need examination. The following file types are those of interest however there are several more depending on the type of virtual machine and the software that was used to create it.

- .vmsn - Virtual Memory Snapshot file, file that containing the memory data at the time of the snap shot.
- .vmx - Configuration file, could also show if external media has been used
- .vmdk - Raw Disk data of the appropriate disk format.
- .log – logs of how it booted

Several tools were tested to examine the virtual machine. As per sound forensics procedures, a copy of the image was taken, a hash value calculated, and the remainder of the investigation was done on the image copy (dd image).

The VMDK files attributes were set to read only and these were used to attempt to start the virtual machine. This failed with a "not enough permissions" error. Then, forensic image mounting software, Mount Image Pro (Get Data, 2008) was used. This mounted both the dd image and the VMDK but provided little in the way of information except that it identified that it was a Linux file system. Then, Forensic Toolkit -FTK Imager from AccessData Corp was used which read the files and offered the best drill down into the file system allowing recovery of deleted files and stored file, together with the date stamps on documents etc. FTK however would not load the VMDK files as an image but would load it as a single file. This did not yield much evidence. The extract below gives some idea of when the Metasploit framework was last initialized.

> Query: "framework" <ASCII/Unicode, Case Insensitive> -- 6794 Hits in 125 Files, 26 Hits -- [Other Linux 2.6.x kernel-000001.vmdk_039] C:\BACKTRACK3\Other Linux 2.6.x kernel-000001.vmdk_039, Offset 170F055 (24178773) -- Security <<Framework>> v1.0.0 initialized. Aug 26 06:48:38 (none) kernel: CPU: L1 I cache: 32K.

This sequence required more research. The failure of this software may have been due to the restraints of the demonstration version of the software. Also, the files contained in the slack space of the image were visible. VMware also has a utility to mount virtual machines without the use of player or server software. With Vmware-Mount installed on a Windows machine it was possible to mount the image. Unfortunately, the software was not able to read the image as it did not recognize the images file system. Subsequently, Window's wanted to format it. When the image was installed and mounted on a Linux environment, the file system was visible and the files could be extracted.

With the correct tools it is possible to investigate the virtual machine by way of the VMDK files. However, it was not possible to ascertain when any of the tools within the environment had been used. The discovery of a date and time stamp would have to be performed on the virtual machine files themselves outside of the virtual environment, and this would provide the time and date it was last used. However, depending on the detection methods and logs acquired, it may be difficult to prove anything happened except on the day and time recorded on the file. However, this data may also have been tampered with before deletion.

ANTI-FORENSIC

Anti-Forensics is a term which has different connotations to different people or groups. It has been described as "attempts to negatively affect the existence, amount, and/or quality of evidence from a crime scene, or make the examination of evidence difficult or impossible to conduct" (Rogers, 2006). Further it has been described as "application of the scientific method to digital media in order to invalidate factual information for judicial review" (Brown & Lui, 2006). In consideration of the scenario posed in this paper, it is useful to consider Rogers' definition - attempting in the process to discover if a virtual machine and the tools contained inside it, can be used to hinder or cover up any evidence of what has been done or what tools have been used.

According to Rogers (2006) anti-forensic tools come in four varieties; data hiding, artefact wiping, trail obfuscation and forensic tool direct attacks. These categories are fairly self explanatory, data hiding concerns it self with hiding data i.e. stenography, hiding data in slack space etc; artefact wiping would be deleting data either by normal methods or by using propriety tools to wipe the data. Trail obfuscation and attacks against forensic tools is the main concern within the context of this paper. For instance, file dates being changed, log files being altered and the fact that using the *.vmdk it was not possible to readily access or recover data without extra tools. This can itself be considered an attack of forensic tools or at the least trail obfuscation. There are some tools that could be used from within the virtual environment to hide data or shred data so that it could not be recovered and therefore be the ultimate anti-forensic tool. As Kessler (2007) noted, "cryptography is the ultimate anti-forensic tool, and has and will continue to make digital investigations difficult or impossible". Any type of encryption, PGP, SSH SSL, whether of a disk, a file or connection will hamper or kill off an investigation. Tools like Metasploit's Sam Juicer or Timestomp can only be used from the Windows host system and could be used on the virtual disk. However, this does not make the virtual machine an anti-forensic tool on its own, it would still require input from the host system.

CONCLUSION

From this investigation it can be concluded that a virtual environment of an operating system with correct tools can indeed circumvent the security of the organization. What occurs after the virtual machine is loaded depends largely on the types of security and policing that is in place. Organizations are certainly at risk from an attack from inside, not only attacks but infections of other machines through installation of rogue software on the virtual machine. Forensic extraction of useful information needs more research. In this case it was possible to see detect that a virtual machine had been used to circumvent the security and some useful files were collected to prove what type of information the attacker was gathering. Dates and solid evidence of what exact data had been viewed was more difficult to collect, however the amount of data that was collected might be sufficient enough for an employee to be dismissed on the grounds of misconduct. The evidence collected however may not have been of a legal quality or quantity.

Further, it should be considered that virtual machine could be used as an anti-forensic tool. Particularly, this would have application in slowing the forensic process down, although the virtual machine can certainly contain tools that could stall an investigation and can get round most of the security of the organization. Further, it may prove impossible for an investigator to determine which tools were used, however this would still only slow an investigation. The fact that something has to be installed unnoticed, or unauthorised by the organisation, is what should be of concern for an organisation. Thus, organisations should be aware of this and should take measures to find or stop any unauthorised installs. This subsequently leads to the emerging problem of ThinApps or portable Apps that can be brought in on small portable devices, such as USB sticks, and run straight from the device without needing to be installed and leaving no trace of ever having touched the machine.

REFERENCES:

AccessData.(2008). FTK Imager. Retrieved September 9, 2008 from http://www.accessdata.com/forensictoolkit.html.

Brown F, & Liu, V.(2006). *Bleeding-Edge Anti-Forensics.* Presentation at InfoSec World 2006. Retrieved September 6, 2008 from http://www.stachliu.com/research_conferences.html/ .

Designer, S. (2008). *John the Ripper*. Retrieved September 1, 2008 from http://www.openwall.com/john/.

GetData Pty Ltd (2008). *Mount Image Pro*. Retrieved September 1, 2008 from http://www.mountimage.com/.

Kessler, G. C. (2007). *Anti-Forensics and the Digital Investigator*. Unpublished Paper. Champlain College Burlington, VT, USA. Edith Cowan University.

Lui, V. & Stach, P. (2006). Defeating Forensic Analysis. CEIC Lecture, 6-10. http://www.metasploit.com/data/antiforensics/CEIC2006-Defeating_Forensic_Analysis.pdf

Lyon, G. (2008). *Nmap*. Retrieved September 1, 2008 from http://nmap.org.

Metsploit LLC. (2008). *Metssploit Antiforensic homempage*. Retrieved September 1, 2008 from http://www.metasploit.com/research/projects/antiforensics/.

Metasploit. (2008). *Metasploit Framework 3 (Version 3).* Retrieved September 1, 2008 from http://www.metasploit.com/framework/.

Metasploit. (2005). *Remote Rogue Network DetectionTechniques and Implementations*. Retrieved September 4, 2008 from http://metasploit.com/research/projects/rogue_network/.

Openwall Project. (n.d.). *Jack the ripper password cracker.* Retrieved September 7, 2008 from http://www.openwall.com/john/.

Ornaghi, A. (2003). *Man in the middle attacks Demos. Blackhat* [Online Document]. Retrieved September 6, 2008 from http://www.blackhat.com/presentations/bh-usa-03/bh-usa-03-ornaghi-valleri.pdf.

RemoteExploit. (2003). *BACKTRACK 3 (Version 3).* Retrieved September 1, 2008 from http://Remote-Exploit.org.

Rogers, M. (2006). *Anti-Forensics: The Coming Wave in Digital Forensics.* Retrieved September 7, 2008 from http://www.cerias.purdue.edu/symposium/2006/materials/pdfs/antiforensics.pdf.

Ruykhaver, J. (23 January 2008). *What are the security risks of virtualization*. Daily Wire. Retrieved September 1, 2008 from…

Whalen, S.. (2001,). *An Introduction to ARP Spoofing", Node99* [Online Document]. Retrieved September 5, 2008 from http://www.node99.org/projects/arpspoof/

Song, D. (1999). *Dniff*. Retrieved September 1, 2008 from http://www.monkey.org/~dugsong/dsniff/.

Spangler, R. (2003). *Packet Sniffing on Layer 2 Switched Local Area Networks.* Packetwatch Research. Retrieved September 5, 2008 from http://www.packetwatch.net/.

Valleri, M. (2005). Ettercap. Retrieved September 1, 2008 from http://ettercap.sourceforge.net/.

Vmware Inc. (2007). *Virtulization White Paper*. Journal. Retrieved September 1, 2008 from http://www.vmware.com/pdf/virtualization.pdf.

Vmware Inc. (2008). Retrieved September 1, 2008 from http://www.vmware.com.

COPYRIGHT