Pitfalls of Devising a Security Policy in Virtualized Hosts

Dennis C. Guster[1], Olivia F. Lee[2], Dustin C. Rogers[3]
[1]Department of Information System
St. Cloud State University
St. Cloud, USA
[2]School of Business
Pacific Lutheran University
Tacoma, Washington 98447
USA
[3]BCRL
St. Cloud State University
St. Cloud, USA
{dcguster,rodu0601}@stcloudstate.edu, olee@plu.edu

ABSTRACT: *The paper provides an overview of three common virtualization threats that have been observed in log files in the authors' network and suggests solutions to mitigate those security vulnerabilities. The solutions offered have been implemented on a network with over 200 hosts 40 of which are virtualized.*

Keywords: Virtualization, Security Policy, Log Disaster Recovery

Received: 14 March 2011, Revised 17 April 2011, Accepted 21 April 2011

1. Introduction

The purpose of this paper is to present and discuss several security threats associated with virtual hosts that have been identified in the authors' log files that effect virtualized hosts. Virtualization creates a new layer of abstraction that often complicates an organization's security strategy. This is because information technology personnel tend to focus on hardware issues such as protecting the computer hosts but not the virtual zones that are created within the host [1]. A recent Gartner Group survey reveals that by 2012 about 60% of virtualized data centers are expected to be less secured than they are now [2]. Many experts argue that over the next decade almost all major data centers will take advantage of virtualization [3]. Notwithstanding that some features of virtualization can enhance security, many unknown threats are troublesome and dealing with the unknowns can be risky [4]. A popular way to cope with complexities of virtualization involves extensive automation of processes [5]. While automation offers some degree of promise, it requires a well thought-out and effective security policy in place to drive the automation [6].

Virtualization allows multiple virtual machines to run on a single physical machine, with each virtual machine sharing the resources of that one physical computer across multiple environments. Different virtual machines can run different operating systems and multiple applications on the same physical computer. As many organizations are leaping aboard the virtualization bandwagon now [3, 4], they need the management tools to run those machines and support a wide selection of applications and infrastructure services their businesses depend on. The ability to build a virtual infrastructure with a proven platform that scales across hundreds of interconnected physical computers and storage devices is critical to successful management

of any data center. It requires careful consideration of assigning servers, storage space, and network bandwidth to every application. A well-developed virtualization plan using an effective security protocol can increase service availability and reduce the physical complexity of the design [5]. To begin, we present an overview on three major concerns: physical host configurations, backup strategy and side-channel attacks, an increasingly common threat to virtualized zones [7].

2. Background Information

Virtualization allows multiple virtual machines to run on a single physical machine, with each virtual machine sharing the resources of that one physical computer across multiple environments. Different virtual machines can run different operating systems and multiple applications on the same physical computer. As many organizations are leaping aboard the virtualization bandwagon now [3, 4], they need the management tools to run those machines and support a wide selection of applications and infrastructure services their businesses depend on. The ability to build a virtual infrastructure with a proven platform that scales across hundreds of interconnected physical computers and storage devices is critical to successful management of any data center. It requires careful consideration in assigning servers, storage space, and network bandwidth to every application. A well-developed virtualization plan using an effective security protocol can increase service availability and reduce the physical complexity of the design [5]. To begin, we present an overview on two major concerns: physical host configurations and side-channel attacks, an increasingly common threat to virtualized zones [1].

3. Physical Infrastructure

A virtual machine in a given physical host acts as a policy interpreter rather than a resource provider. Configuring a virtual machine to fulfill two missions (security reference monitor and resource emulator) often limits the effectiveness of a virtualized solution. This is because its "double-duty" lessens performance and creates complexity within the virtualization layers. As a result, increased security creates complexity with sophisticated codes that sometimes invoke trust issues from a user's stand point [8]. Devising a strategy to implement these codes in the physical world can be challenging. The physical host configuration strategy consists of taking a physical machine and deciding how it will be subdivided into logical zones. For example, one of the zones could contain a dynamic host configuration protocol (DHCP) server. That zone in effect becomes the policy implementer for how workstations are allocated temporary IP addresses. However, it has no physical resources of its own and is dependent on how the main zone (virtualization supervisor software) allocates physical resources. This is typically not a problem unless multiple virtual zones are all running at a high utilization level.

Therefore, the primary concern in regard to physical host configuration is to balance the number of zones with available resources. A configuration with 10 zones, for example, would run with reasonable response time but as more zones are added, the response time may not be acceptable. When each zone is isolated, trustworthiness of tenant becomes an important concern since sharing hardware is risky. A major threat can occur when the network interface card (NIC) is placed into promiscuous mode which traps all network traffic on the physical host, allowing information be retrieved from any zone.

Many studies have been conducted to address security and trust policy especially on integrating trust through a virtualization strategy to enhance security within a computer grid [9, 10, 11, 12]. The concept of modularity is used to break complex problems into subparts hierarchically, and apply structured layers to virtualized hosts. The architecture's hypervisor and all virtual machines are used to split policies into multiple layers. A main advantage is to enforce security while not increasing the operation overhead within the overall security strategy on the total system [13].

4. Side-Channel Attacks

4.1 Service via software applications is gaining popularity in today's business operations. Increasingly, more and more applications are delivered to the client via the internet. Unlike a desktop application, a web application is split into browser-side and server-side components. A subset of the application's internal information flows is inevitably exposed on the network. A side channel attack is referred to any attack based on obtained information from the physical implementation of a cryptosystem, when central processing unit (CPU) and/or memory (RAM) are shared between the victim and attacker [14].

To illustrate, imagine a row of houses on a street that share a common water pipe. When an individual living in the third house turns on the shower, the person in the fourth house notices the decrease in water pressure. Therefore, the former can deduce information about his next door neighbor, in this case, when s/he takes a shower. In a recent study, Root Labs security

consulting firm investigated the formidability of three types of side channel attacks on cryptographic software.

The rise of side-channel attacks has gone through various changes. In the past, a malicious attacker listened remotely to monitor interested variables and their timing jitter often disrupted the data flow, causing inaccurate results in their attempts to launch side channel attacks. This limitation can be overcome by the attacker if they can get onto a physical system but in a separate virtual zone. When the attacker is on the same physical host, but in a different zone the attacker is then on the same motherboard and timing jitter is no longer a problem. In the current virtualized era, one can no longer assume that the attackers are remote. As newer and advanced applications are designed, their inherent flexibility allows resources to be effectively shared across a physical system which creates more possibilities for greater precision measurements which could be used in side channels attacks [15]. Henceforth, these attacks that were once esoteric are becoming more apparent in today's cloud computing environment.

Despite encryption, side-channel attacks present a serious threat to users' privacy as a result of information leaks. All side channel attacks have the same basic premise. The hackers analyze gathered data that was never meant to be valuable, as the data is generally a side effect of some part of the computation process [16]. The hackers are knowledgeable about what data are being requested such as data formats, timing relationships, and user requirements. The general classes of side-channel attacks include timing attack, power monitoring attack, radiation monitoring, acoustic cryptanalysis, differential faulty analysis, and telescope or distance observation. Because one of the primary attacks identified from the author's autonomous was related to timing and power monitoring a brief review of how these types of attacks takes place will follow.

5. Side-channel Attacks Based on Timing or Power Monitoring

Most online accounts require a secret question to facilitate password recovery. The correct answer to which is supposed to validate that a given user is the legitimate owner of that account. The problem lies in the fact that a secret question is generally not too secret and can be easily guessed by hackers via social engineering. Under this type of side channel attack, the attacker repeatedly observes the processing time of the same cryptographic function.

One timing based example involves decoding of a 128bit key. This variation is deemed "Keeping the correct answer secret" [15]. Specifically, the key was compromised by forcing consecutive binary comparisons of the crafted hashed message authentication code (HMAC) with the correct HMAC. The attacker keeps sending incorrect HMACs to the target victim's system. Since bitwise comparisons are used to verify HMAC, the attacker merely records the time it takes to be rejected. With each attempt, the attacker can verify a few more bits since the comparison will take a little longer for each consecutively correct bit. This method allows one to be able to deduce the keys in different high-level languages including C++, Java, and Python.

A second common variation of side-channel attacks involves monitoring the program's footprint. When an interrupt system program runs periodically in a cache-based computer, a short cache-reload transient occurs each time the interrupt program is invoked [17]. The portion of a cache used by a program in effect provides footprints of the various programs in the cache. A program footprint is defined to be a set of lines in the cache in active mode within the program. Based on manipulating and observing memory cache registers, these lines are called by either the victim (evict and time attack), or attacker (prime and probe attack) [18]. The characteristics of the reload transient system process depend upon the cache size and on the sizes of the footprints for each of the competing programs. This is another security threat since every virtual zone has legitimate access to the cache. By reading the contents of the cache, an attacker could extract a portion of some sensitive information and obtain data that could be used to defeat some security mechanism.

A third common variation involving side-channel attacks has been termed "Which way did he go?" [15]. This form of timing/ monitoring attack is similar to an older method where the attacker takes advantage of how central processing units (CPU) compute numbers. Certain types of binary arithmetic occur only if the value is "1". The attacker merely records multiple samples of the amount of time it takes to encrypt the same plaintext [19]. Longer computations at a specific point mean that the key held a value of "1" therein. In a "Which way did he go?" attack, specific CPU records known as "branch prediction units" are analyzed by the attacker through observing timing latencies.

The impact of this type of attack is substantial when examining the architecture of a digital computer. Being able to decipher a "1" provides the attacker with half of the possible bits and if the timing interval is known the other bit types have to be "0".

Therefore, this type of attack can provide all information that might be going in and out of the CPU to a malicious hacker. In a virtual world when sharing resources has become an inevitable common practice, it is disturbing to depend on the integrity of users in other zones to act ethically in and retrieving accessing information.

Certainly all of these specific types of side channel timing/monitoring attacks illustrate the potential for catastrophic vulnerabilities in virtualized hosts. It is therefore critical that in a virtual world in which physical resource are shared that policy is in place to protect again such attacks.

6. Virtualization Management Strategy

With the rapid adoption of virtualization, there is a great need for a standard way to package and distribute virtual machines. Researchers continually develop cloud computing techniques that efficiently allow for the pooling of disk space and memory to improve the efficiency of CPU cycles. One of the primary objectives behind distributed technology is to designing a platform that is independent, efficient and extensible, where resource utilization and security measures can co-exist and remain hacker-free. Extant studies reported that any policy to prevent side-channel attacks is ineffective without a staff with a high level of technical expertise, a sophisticated hardware infrastructure and software properly configured for a virtual world. Understandably defending a virtual computing world will substantially increase the overhead expenses and operating cost [20]. A simple and cost effective strategy is to modify the computational techniques used in encryption software. This is a simple process applicable to almost any algorithm and may not significantly add to execution time. These algorithms which use additional calculations to mask the characteristics of the real calculations act like a "wrapper" to a specific algorithm such as RSA (for Rivest, Shamir and Adleman which is an algorithm for public-key cryptography). Other related techniques are novel exponentiation algorithms, protected square-and-multiply algorithms, right-to-left counterpart techniques and several protected sliding-window algorithms [21]. While these techniques show promise, vulnerabilities and performance problems still remain major concerns [22].

Because economy of scale and technical capabilities limit what organizations can do to protect their virtual environment, some data centers have started to build their own comprehensive solutions based on a conglomerate of software applications integrated with components from object directory services, the operating system and identity managers. Large businesses with large and well-established IT resources are able to adopt this approach. Small businesses, on the other hand, are unable to do the same, at least in the short term [23].

7. Methodology

This paper presents a series of attacks on virtual resources within an autonomous system of a research laboratory. A series of attacks including, side-channel attacks, were observed in the system's log files. The attacks represent real-time incidents that occurred during a period of several months. It is noteworthy to report that the attacks were not caused by experiments or stimulations generated for research purposes. Rather, they represent random attacks by unknown sources to obtain information by exploiting security loop holes caused by virtualizing hosts. Fortunately, the majority of the attacks occurred while the system was still in research/development mode. As a result of these observations the design was modified several times and the security team is in the process of devising additional tools to protect against timing/monitoring side-channel attacks.

The autonomous system of the research laboratory has a clearly defined routing policy that defines access to over 200 hosts, approximately 40 of which are housed in virtual zones. This research laboratory provides resources to support instruction within the university, resource to support graduate student/faculty research and high performance computing resources used in sponsored research. The system is physically located in a mid-western university and it was in designed to reduce the: (1) number of physical hosts, (2) amount of physical space required to store the hosts, (3) amount of electricity required to run and cool the hosts, and (4) complexity of the design as a means to save personnel costs for regular maintenance. Prior to virtualization, 10 physical hosts were required to support the production related services. This was reduced to one physical host with 10 virtual zones. Each zone represents one service such as domain name service (DNS) or dynamic host configuration protocol (DHCP). Because these services are mission critical three replicas were configured to provide fault tolerance and load balancing. The main production physical computer and Replica number 1 are housed together in the same equipment room. Replica number 2 is housed in a different building on campus about two blocks away and the third replica is housed in another city about 500 miles away. These computing resources are monitored and managed by about five technical staff.

8. Problem Identification

The unexpected attacks on the research laboratory's autonomous systems provided the researchers an opportunity to identify real-world design deficiencies and side-channel attacks. In this section, the paper outlines three common attack scenarios the author's observed related to their virtualized configurations.

8.1 Problem 1: Replication, Fault Tolerance and Fail-over Procedures

In a modern virtualized environment where software applications are deployed, direct communication channels across the host system's motherboard are used to optimize data flow. The traffic going through the internet cloud is the wide area network (WAN) traffic. In this type of virtualized setting, there will always be replication traffic limited by the available bandwidth within the WAN. The traffic internal to the host is moving across the motherboard at a high rate of speed. The replication/fault tolerance/fail-over can be accomplished by duplicating the virtual environment on a remote or second host or the replica. If the original host expires, then the entire virtualized environment will fail over to the second host. When configured properly, communication between the virtual machines will continue over the second host's motherboard.

In the event when only one system fails on the original host, only that system will fail-over to the remote or the second host. This situation means that some traffic is still traversing the first system's motherboard, while some of the traffic must rely on a slower media either a LAN or WAN connection. The log files indicated that in a one week period from 9/27/09 to 10/05/09 that this problem occurred eight times. A specific example of concern occurs when only the domain name service (DNS) is configured to fail-over. The dynamic host configuration protocol (DHCP) tends to be site-specific so it may not be logical to fail it over. Hence, some traffic is traversing through a slow WAN via the internet cloud. However, when traffic is sent across the motherboard only improved performance is observed especially if compared to a WAN. Since connections across a WAN coupled with the security overhead can significantly slow transmission time, network services can immediately be brought to a standstill, effectively crippling the whole autonomous system.

There are three possible ways to solve problem.

8.1.1 ***Understanding service dependencies***. The first solution involves writing a policy that takes into account dependencies when services are transferred. For example, When the DNS service is transferred, a database server that takes advantage of distributed resources should be transferred as well to allow the DNS to find those resources. In other words, services that intercommunicate with one another need to be on the same replica to maintain adequate communication speed. To combat this problem a fail-over state in which all services except DHCP are configured to fail-over to the tertiary hypervisor can be used. In this scenario the traffic traversing through the internet cloud would be viewed as slow, WAN traffic while the traffic being transferred across the motherboard and would run rather quickly.

8.1.2 ***Define optimal utilization***. The second solution involves writing a policy that defines optimal utilization of "host-only" network traffic so that LAN/WAN speeds are more equivalent. Essentially, this policy would state that bandwidth across the motherboard would be regulated by policy to be limited to the same speed as the WAN links. Traffic flows would always run at the same speed and maintain the security boundary that motherboard-only traffic flows would provide. This solution will equalize access opportunity to the media but would result in reduced data transmission speeds.

8.1.3 ***Allow only necessary services***. The third solution involves writing a policy that defines that only necessary services should be running during any fail-over. This is a short term solution that may slow down the recovery of those services that are deemed not crucial. Since the WAN link becomes more crucial during a fail-over state, it is advisable to eliminate unnecessary WAN-type services.

Because of the moderate volume of data associated with the system it made sense to pursue a total fail over policy. In other words if one zone fails all zones are migrated to a healthy replica. This methodology has significantly reduced the incidence of this problem.

8.2 Problem 2: Backup Problem

A common method in use today is to simply back up the entire host system as a single entity. By backing up the host system, each of the "guest's" virtual machines is backed up as part of one contiguous file. This process is problematic because if a small portion of the file becomes corrupted then the entire file in effect becomes corrupted, thereby incurring the loss of the

virtual machine's backup copy. During the period of 12/21/09 to 6/13/10 this problem occurred ten times.

To solve this backup problem, a policy was devised that defines the extent to which virtualized machines are backed-up. Because the virtual zone can be viewed from two different perspectives: an independent entity with its own data and a subpart of the physical host, backing up the file system in its natural hierarchy, and as a single file within a virtualized host would be a good practice. After changing the policy to ensure each zone was backed up independently the number of incidents was greatly reduced.

8.3 Problem 3: Danger Associated with Shared Resources

Data centers taking advantage of virtualization often outsource institutional hardware requirements to other organizations. When one of those virtual hosts resides on the same physical host as some other company's virtual host, it creates vulnerability for side-channel attack when resources such as central processing unit (CPU) and/or memory (RAM) are shared between the victim and attacker. Since anyone with a virtual host on a physical machine has access to the hardware, a hacker can monitor traffic moving across the main-bus and intercept another users' sensitive data. In addition, a virtual machine on the same physical host could create a denial of service attack just by running a higher workload. Figure 1 illustrates how a virtual host might be (mis)configured to be highly vulnerable. In this example, virtual machines are only configured to use one processor core that regrettably the web server clearly shares with potential attacker. Memory or RAM is also shared with the attacker and thus could be exposed to be "mined" for sensitive data. The authors' log files in fact indicated that there were 10 incidents in the period from 12/20/9 to 2/21/10.

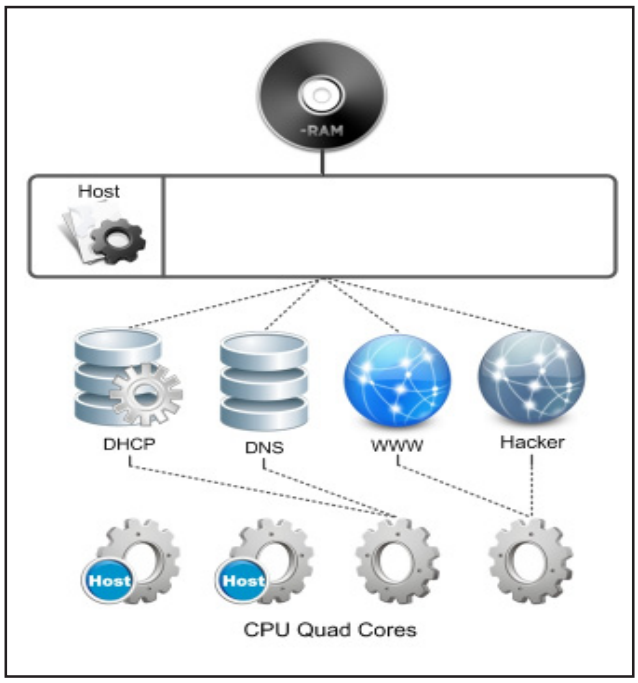

Figure 1. Vulnerable Hardware Design

There are two possible solutions to address problem 3.

8.3.1 ***Restrict outsourced activities***. The first solution involves writing a policy that clearly defines the acceptable characteristics of a vendor that will provide safe outsourced virtual resources. Certainly it is critical that virtualization centers follow strict screening procedures of the clients they are willing to accept. Organizations need to write policy that demands that outsourcing of virtualized systems only be allowed on hardware that is dedicated only to their use and therefore, other clients of the virtualization center will not share the same allocated hardware. Therefore, select data centers should feature: (1) professional-grade hypervisor for their virtualization host applications; (2) operating systems that employ symmetric multi-processing; (3) assigned multiple cores; (4) RAM dynamically allocated in a discrete fashion; and (5) memory isolated from other virtual machines.

Figure 1 depicts a vendor configuration that would be considered vulnerable because virtual zones share memory and CPU resources. Figure 2 shows a second configuration that is less vulnerable. In this improved design all VMs and the Host (H) are configured to use symmetric multiprocessing (SMP), thus they all use two processor cores, and no two machines share the same two cores. Also, memory (RAM) is dedicated to each machine, so no one VM, or the host, shares memory with the potential attacker.

8.3.2 ***Limit failed attempts***. The second solution involves writing a policy that defines the maximum number of failed attempts to access cryptographically protected services such as (HTTPS), or secure shell. Once the failed maximum has been reached, a particular IP address associated with that activity would be locked out. However if the attacker is controlling a botnet or spoofing IP addresses as an extension, this hacker is able to make many attempts on the cryptographic service. Such policy needs to cover the CPU level and limit the attempts based on the process or parent process ID. This was the solution the authors' selected.

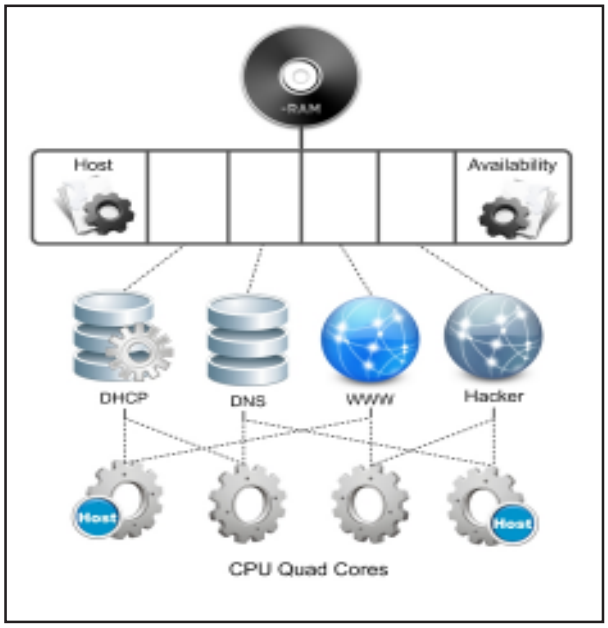

Figure 2. Safer Virtualized Host Design

9. Discussion

In order to effectively operate a data center with limited resources it was necessary for the authors to embrace virtualization. While virtualization was successful in reducing the footprint of the data center it created a number of security vulnerabilities not present with the traditional physical host model. In other words, virtualization offers attractive security isolation, but it also creates vulnerabilities when hardware is shared. These vulnerabilities are backup/fault tolerance considerations, virtualization (design) configuration, equal access configurations and side-channel attacks typically related to timing/monitoring considerations. These vulnerabilities might appear subtle in nature but their identification and diagnosis requires a fairly high level of sophistication in regard to hardware design and log file analysis. Based on a series of initial unexpected design deficiencies and side-channel attacks on an autonomous system supporting a research laboratory, the authors presented vulnerability levels and practical solutions to address the three identified problems. These solutions are particularly noteworthy for newly virtualized data centers. Because the corrective measures related to timing/monitoring attacks need to occur in real time to be effective. Therefore, automating the process is crucial.

This need to automate is particularly applicable to problem 3 in that the implementation of the policy defines the maximum number of failed attempts allowed to access services that are cryptographically protected. To be effective this process needs to be automated because thousands of brute force attempts to break an encrypted password can take place in one second. A strategy to prevent this is well developed in regard to attacks coming from a network. That strategy simply blocks the attacking IP address in some form on the network firewall. This strategy can be simplified as follows: a monitoring process once detecting such a problem sends a request to the firewall to block the offending IP address and that change can take place in a matter of

milliseconds. In the case of virtualization those attacks may originate from the physical host's mainbus and therefore this same logic must be applied in which a monitoring process (i.e., analogous to the network firewall) identifies brute force attacks and kills the appropriate offending processes and/or their parent process.

Because virtualization allows multiple virtual machines to run on a single physical machine, each virtual machine shares the resources of that one physical machine resulting in infrastructure design flexibility. Different virtual machines can run different operating systems and multiple applications on the same physical computer. As many organizations are leaping aboard the virtualization bandwagon now [5, 6], they need the management tools to run those machines and support a wide selection of applications and infrastructure services their businesses depend on. The ability to build a virtual infrastructure with a proven platform that scales across hundreds of interconnected physical computers and storage devices is critical to successful management of any data center. It requires careful consideration of assigning servers, storage space, and network bandwidth to every application. A well developed virtualization plan using an effective security protocol can increase service availability and reduce the physical complexity of the design [1]. However, as previously stated sound policy development is critical to ensure success.

10. Future Research Directions

This paper aims to shed light on managing attacks to hosts within a virtualized environment. The practical solutions presented here can be enhanced in several ways through future research. To limit its scope of coverage, this paper does not cover other aspects of side-channel attacks such as TEMPEST, acoustic and differential fault analysis. Based on observations gleaned herein there are two promising avenues for future research. On a virtualization strategy level, more work needs to be focused on devising a policy to deal with the unavailability of comprehensive prevention tools. On a technical level, the use of "wrapper" algorithms to mask the physical signal modulations occurring on the motherboard (i.e., allows the hacker to extract the 1's and 0's) is proving successful. However, like most protective tools their value appears to be dynamic in nature, therefore their effectiveness needs to be monitored and updated as hackers gain sophistication in defeating them. In addition, at the present time off shelf software is limited in dealing with side-channel attacks. The development of such products will be crucial for smaller organizations with limited IT resources and expertise to adopt virtualization. Similar strategies to those presented in this paper can also be applied to brute force attacks on the mainbus level. Hence, research related to the design of a monitoring process that will prevent these brute force processes requires further exploration. In summary, virtualization does offer numerous benefits, but there are numerous vulnerabilities that need to be addressed particularly in situations in which an organization with limited IT resources is using outsourced virtual machines.

11. Acknowledgement

The authors gratefully thank Emily Goenner for her editorial advice of this manuscript.

References

[1] Guster, D.C., Hemminger, C., and Krzenski, S.(2009). Using virtualization to reduce data center infrastructure and promote green computing. *International Journal of Business Research* 9(6), 133-139.

[2] Help Net Security. Six common virtualization security risks and how to combat them. http://www.net-security.org/secworld.php?id=9023.

[3] Siebenlist, F. (2009). Challenges and opportunities for virtualized security in the clouds. *In*: Proceedings of the 14th ACM Symposium on access control models and technologies, Stresa, Italy, 1-2.

[4] Vaughan-Nichols, S.(2008). Virtualization sparks security concerns, *Computer,* 41(8) 13-15.

[5] Cabuk, S., Dalton, C., Eriksson K., Kuhlmann, D., Ramasamy, H., Ramunno, G., Sadeqhi, A., Schunter, M., Stuble, C (2010). Toward automated security policy enforcement in multi-tenant virtual data centers, *Journal of Computer Security*, 18(1) 89-121.

[6] Clancy, H. Tech watch: Security pros want strong policy for virtualization (2011).http://searchitchannel.techtarget.com/news/1357537/Tech-Watch-Security-pros-want-strong-policy-for-virtualization.

[7] Christodorescu, M., Sailer, R., Schales, D., Sgandurra, D., Zamboni, D. (2009). Cloud security is not (Just) virtualization security, *In:* Proceedings of the 16th ACM Conference on Computer and communications Security, Chicago, Il, 97-101.

[8] Bratus, S., Locasto, M., Ramaswamy, A., Smith, S. (2008). Traps, events, emulation and enforcement: Managing the yin and yang of virtualization-based security, *In:* Proceedings of the 15th ACM Conference on Computer and communications security, George Mason University, VA, ACM Press, 49-58.

[9] Abbadi, I.M. (2009). Secure information sharing for grid computing. *Security and Communication Networks* 2 (6) 144-151.

[10] Lohr, H., Ramasamy H., Sadeghi, A., Schulz, S., Schunter, M., Stuble, C. (2007). Enhancing grid security using trusted virtualization, Lecture Notes in Computer Science, 4610, 372-384.

[11] Wang, H. (2008). A novel trust enhanced grid authentication mechanism. Wuhan University Journal of Natural Sciences 13(5) 528-532.

[12] Wang, H. (2010). Trust asymmetry in grid authentication, *Wuhan University Journal of Natural Sciences*, 15(3) 201-204.

[13] Payne, B., Sailer, R., Caceres, R., Perez, R., Lee, W. (2007). A layered approach to simplified access control in virtualized systems, *Operating Systems Review*, 41(3) 12–19.

[14] Bar-El, H.(2010). What are side channel attacks? HBarel.com. http://www.hbarel.com/Misc/side_channel_attacks.html

[15] Lawson, N. (2009). Side-channel attacks on cryptographic software, *IEEE Security & Privacy* 7(6). 65-68.

[16] Zhou, Y., Feng, D. (2005). Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing. Cryptology ePrint Archive, Report 2005/388, October 2005, 1-34.

[17] Dwoskin, J., Gomathisankaran, M., Lee, R. (2009). A framework for testing hardware-software security architectures. Princeton University Department of electrical engineering technical report CE-L2009-001, June.

[18] Osvik, A. Shamir, A., Tromer, E. (2006). Cache attacks and countermeasures: The case of AES, Topics in Cryptology, CT-RSA 2006, LNCS 3860, Springer, 1–20.

[19] Aciiçmez, O., Koç C. and Seifert, J. (2006). On the power of simple branch prediction analysis. *In:* Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security. ACM Press, 312–320.

[20] Tiri, K. (2007). Side-channel attack pitfalls. *In:* Proceedings of the 44th annual Design Automation Conference San Diego, California 15-20.

[21] Benoit, C., Ciet, M., Joye, M. (2004). Low-cost solutions for preventing simple side-channel analysis: Side channel atomicity, *IEEE Transactions Computers*, 53(6) 760-768.

[22] Kim, H. and Quisquater, J. (2007). How can we overcome both side channel analysis and fault attacks on RSA CRT? *In:* Workshop on fault diagnosis and tolerance in cryptography FDTC, 21-29.

[23] Herrick, R. Jr. (2010). Is virtualization and security as unsettling as it sounds. http://software.intel.com/en-us/blogs/2010/05/18/is-virtualization-and-security-as-unsettling-as-it-sounds/.

Author Biography

Dr. Dennis C. Guster is a Professor of Information Systems and Director of the Business Computing Research Laboratory at St. Cloud State University, MN, U.S. His research interests include network design, network performance analysis and computer network security. Dennis has over 25 years of teaching experience in higher education and has served as a consultant and provided industry training to organizations such as Compaq, NASA, DISA, USAF, Motorola, and ATT. Undertaking various sponsored research projects, Dennis has published his works in computer networking/security journals.

Dr. Olivia F. Lee is a visiting assistant professor of Marketing at Pacific Lutheran University. She has worked as an operation manager at two university hospitals and as a senior e-business market analyst in a business-to-business company prior to her academic career. Her research work focuses on technology practice in business environment, health care and service organization, and business resilience strategy. She has published her work in marketing, management, and information technology journals.

Dustin Rogers is the security officer for the Business Computing Research Laboratory and network administrator for the Computer Networking and Applications program at St. Cloud State University, where he is a graduate student majoring in Information Technology Security. His areas of interest in security include information assurance, distributed cryptography, and intrusion prevention.