Survey of Security in Type 1 Hypervisors

Brennon York
boyork@indiana.edu

Abstract

With distributed systems becoming more prevalent in modern computing one can witness a correlation between the exponential rise of security necessity and the linear increase in use and integration. This paper intends to provide a brief survey of those security practices and methodologies put in place by production Type 1 Hypervisors. All data was collected through software documentation provided by the corresponding manufacturer and thus take everything as ground truth. This survey does not cover penetration testing or any capability related to attacks with malicious intent (e.g. side channel attacks). This study focuses its efforts on three leading Hypervisor distributions: Xen, VMware ESX(i), and Hyper-V.

Introduction

Hypervisors, also known as Virtual Machine Monitors (VMM's), are applications that allow for the ability to run multiple Operating Systems (OS's) on one or more machines. Just as an Operating System's job is to manage the local resources for a given machine so is the job for the Hypervisor across multiple machines. Typically these Hypervisors will sit on top of a physical computing cluster and provide a layer of transparency to each Operating System it is hosting. This grants each OS supported a given subset of the total physical resources available to the underlying Hypervisor. These types of Hypervisors that sit directly upon the physical components are referred to as Type 1 Hypervisors. Type 2 Hypervisors are differentiated in that they sit above an Operating System, which sits on top of the physical nodes, effectively one level higher than that of Type 1. This survey does not cover Type 2 Hypervisors as their security models and specifications are vastly different from their Type 1 counterparts primarily because of the additional level of abstraction over the OS. This study focuses on the three most commonly used Type 1 Hypervisor distributions currently running in academic and enterprise environments; those being Xen, VMware ESX, and Hyper-V.

Tanenbaum states in [1] that "VMM's will become increasingly important in the context of reliability and security for (distributed) systems. As they allow for the isolation of a complete application and its environment, a failure caused by an error or security attack need no longer affect a complete machine." While this statement holds true in most cases, it does not accurately portray the need for security pertaining to the VMM as a whole. These Hypervisors are essentially the 'command and control' centers for each individual Operating System hosted. This implies that if there is a breach in security at a VMM level, then there is a possibility that the attacker could gain access or information about every Operating System running on the given VMM. These attacks could come from the Operating System being supported by the Hypervisor if there are vulnerabilities lying dormant. Attack vectors such as this effectively mitigate Tanenbaum's statement as his security model only encompasses attacks enclosed within a given supported Operating System. It is this minimized view of security that provides the inspiration for the work presented.

Xen Hypervisor

Xen began as an open source project that originated from the University of Cambridge Computer Laboratory in 2003 and, as of 2010, is now maintained by a community of members. The primary features of the Xen Hypervisor are its wide operational architecture range and ability to emulate nearly all Operating Systems. Currently the Xen system can run on IA-32, x86-64, Itanium, and ARM architectures. It can support all forms of Unix-based Operating Systems and, if utilizing the correct hardware, can also support versions of Microsoft Windows [5]. The special hardware needed pertains to specific virtualization capabilities available in the Intel VT-x [6] and AMD-V [7] processors. The Xen Hypervisor begins by launching a privileged node on startup, denoted as domain 0 or dom0, that controls the operations of each guest OS that launches subsequently, referred to as domain U, or domU. A key aspect of the Xen distribution is that, because of its wide architecture range, it is beginning to be ported to mobile and personal computers (e.g. cell phones, PDAs, tablets, etc.) which furthers the need for this study.

The Xen Hypervisor begins its dive into security with the suggestion that the management domain (dom0) should be held to the utmost security and recommends three additional considerations listed as follows:

1. Run the smallest number of necessary services.
2. Use a firewall to restrict the traffic to the management domain.
3. Do not allow users to access Domain 0.

While those words are true, their actual guide in implementation seems lacking. They move on to describe driver domains. These domains are logical areas within the guest Operating Systems to manage devices. This is done for two main purposes; one, for stability, to be able to restart a device driver without needing to restart an entire machine, and two, to ensure a higher level of security. The latter is considered because on a typical Operating System a device driver will run with root privileges from within the kernel. This is a fatal security breach if Xen were to allow a Guest OS direct access to talk to the dom0 kernel. While the driver domains seem secure there are still many security issues to discuss because of the multitude of hardware compatibilities.

First, most x86-based platforms come without an IOMMU (Input/Output Memory Management Unit) to restrict their reads and writes. This is can lead to a malicious Guest OS which would then have the capability of being able to call arbitrary reads and writes into any memory location. This includes memory that was not allocated to it by the dom0 at the time of launch for the instance. A security breach such as this would grant the end user total control over any Guest OS running as well as the dom0. Because these IOMMU's are not present in many x86-based platforms (and that is one of the most common platforms) it becomes increasingly apparent that this is a major issue. Second, devices that share data buses are susceptible to man-in-the-middle attacks. A domain A owned by some malicious user could watch the data on device A which was assigned to that instance and monitor traffic flow from domain B on device B so long as devices A and B shared a hardware-level data bus connection. Third, domains have the capability of blocking on system calls thus halting all progress to other domains running on that machine. This can be done through device driver interrupts when they are not cleared by the domain that called it when finished. Last, Xen can only restrict access to physical memory at a given granularity. In some cases it is very fine (interrupt lines) while for others it is much larger (I/O memory). In the case of I/O memory its granularity is at the page size. This means that any domain could land on the same page for their I/O memory as one already located there and have the ability to read or write that other domain's data.

For all basic security implementation details of the Xen architecture they refer to the Linux firewall and IPtables commands. For many major Linux distributions (Ubuntu, Fedora, Red Hat) the IPtables commands are the de-facto standard in firewall control, typically built directly into the kernel. Although it is widely popular, it does not provide a complete security implementation, but merely a single wall of many to ensure the principle of defense in depth.

To further clarify security implementations an access control framework was build called sHype, short for secure Hypervisor, created by IBM. Its essential goal is to allow or disallow specific communications and resources on a per domain basis determined by the security policy in place. This is considered a mandatory access control (MAC) policy because everything is handled directly through a single administratively-owned policy. This differs from discretionary access control (DAC) where users define the security policy for their own files. They mention a caveat right from the start though that all security policies handled through sHype are only considered valid so long as the dom0 has not been corrupted. They note the benefits gained through the sHype access control module as:

- Robust workload and resource protection effective against rogue user domains
- Simple, platform- and operating system-independent security policies
- Safety net with minimal performance overhead in case operating system security is missing, does not scale, or fails

The sHype module also grants security for sharing data between domains. It does this in a similar fashion to the SELinux policies by placing security labels onto specified domains and resources. These labels cannot be manipulated by any user-domain and claim it is effective in protecting against compromised and rogue domains. Figure ?? above overviews how the configuration and setup of the sHype system is done.

In Summary, the Xen Hypervisor provides a basic means of security through its domain 0 (dom0). This special host has control over all guest domains that are launched and is granted root privileges to all resources that it manages. To ensure the security of this critical domain the sHype access control module was created to ensure a mandatory access control policy. This strict policy ensures that cross-domain sharing can occur without worries of information leakage. There are small areas of deficiency within the Xen architecture related to the domain drivers. Depending on specific hardware and setup styles the domain drivers can become the point of failure for secure communication to be assured.

VMware ESX / ESXi

VMware ESX is an enterprise-level Hypervisor offered by VMware Inc. through a proprietary license. VMware Inc. offers a multitude of virtualization and emulation software although this paper will focus solely on VMware ESX and VMware ESXi versions. The former constitutes the introduction to Type 1 Hypervisors for VMware Inc. being released in 2001 and providing traditional features [8]. The latter ESXi model built upon the original and reduced its size by 5%, increased Hypervisor management and administrative features, and, most importantly, hardened the security [9]. Like the Xen Hypervisor the ESX and ESXi models leverage a specialized node, called a Service Console, in which a user is granted "Linux-based privileged access" to the kernel [10]. This kernel, known as a vmkernel, allows for three distinct interfaces to between it; one, the Service Console previously mentioned, two, the guest operation systems, and three, the hardware. These interfaces will play a key role in section 4 as the security of the Hypervisor is discussed.

The ESX architecture is part of the underlying set that VMware calls its Infrastructure 3 series. For this series VMware has created a security guide to cover all facets of its implementation and deployment. They begin by explaining that a virtual machine should not be seen as anything different than a typical operating system. As such its security for each guest should be handled just as one would if it were running on its own dedicated hardware. With that they recommend that antivirus, antispyware, and intrusion detection be installed on each guest under the ESX console. Additionally patching should happen occur on each guest, even if powered off, to ensure that all security is up-to-date and minimized the exploitation vectors. The next set of security rules entail the disabling of unused resources or processes. If an operating system is used as a file server then it might not need any web server or DNS capabilities. It is up to the system administrator to ensure that all unnecessary processes are stopped without disrupting the services of the virtual machine. Additionally one wants to disconnect any unnecessary physical devices from the machine. This will negate any additional virtual driver issues that might need to be squashed helping not only to security, but possibly configuration as well.

A feature that VMware ESX(i) provides is that of a template, or snapshot, of a virtual operating system. This grants a major feature of building secure templates and deploying those. This grants each VM instance a baseline level of security without the additional need of configuration for each instance launched. The template system also allows for quick updating. This allows an administrator to quickly update a template with new security software or patches without the need of a complete template rebuild. VMware furthers the template design by stating that each deployed template will grant a specified level of security that can be built upon when customizing the instance to specific needs (e.g. file server, web server). A caveat to this is that one should be extremely careful of what is being installed after the template has been pushed out. A template could have a very high level of security, but if a client were to install an outdated version of Apache then any attempts to ensure secure web traffic could be voided.

The VMware ESX(i) architecture provides an interface for resource management which one can use to control the amount of resources a given guest OS can consume. This is especially important when discussing denial of service attacks from guest instances. Guest A should not be allowed to perform large data calculations which consume 100% of the CPU cycles. This would essentially deny service to the CPU from any other guest operating system sitting on that ESX(i) Hypervisor. This obviously becomes a more interesting task when talking about multi-CPU systems such as high-end servers, clusters, or grids.

A key security practice, that VMware notes should be done on all Infrastructure 3 applications, is the utilization of network hardening. This concept relates heavily to ESX(i) infrastructures because of the way virtual machines typically have their networks set up. A set of virtual machines will sit on top of the Hypervisor with a virtual local area network (VLAN) connecting them. This VLAN allows for any virtual OS to connect or talk to another OS which could prove dangerous if one, or many, of those OSes were malicious. With network hardening one can segregate these guest machines into virtual subnets, individualized VLANs, or even utilize separate physical devices. All of these are "best practice" procedures and do not guarantee a secure network, although they do provide for a greatly reduced set of attack vectors which helps immensely.

Last, VMware begins talking about specifics for hardening the service console. It is crucial to remember that the service console is the primary attack vector because of its immense functionality and control over every virtual operating system hosted on the ESX(i) Hypervisor. They start with firewall setup and configuration practices. The firewall comes preinstalled with a configuration that disables all incoming and outgoing traffic to the service console save for a select few specified ports for which services that must talk to the console run on. Additionally, if one of those services is shut down on the guest instance side, then the service console will recognize that and close its port at well to ensure no ports are needlessly opened. Next, they provide detailed specifications for individual services that run within the service console. All processes running within the service console should obviously be left to a minimum, only running what is needed. All else provide additional attack vectors which therefore reduce security for needless causes. Last, they mention that the service console should only be accessed when absolutely necessary. VMware ESX(i) provide the VI Client and VirtualCenter consoles to manage a vast majority of issues on the guest OS side of things to minimize use of the service console. These instance-specific consoles only grant privileges to administrators of those instances and do not maintain any information regarding other instances possibly running on the Hypervisor.

To conclude, VMware ESX(i) has developed an Enterprise-level hypervisor solution. All security and implementation falls into the hands of their service console which maintains absolute administrative rights to any and all guest instances running. Because of that their documentation specifies many details on how to accurately configure the console for secure use such as firewall setup, process restrictions, and alternative console management (reduced privilege consoles). In addition to hardening the service console VMware ESX(i) infrastructures provide many features to ease building a secure environment such as templates and resource management consoles. They also focus much effort on the hardening of individual guest operating systems running on the ESX(i) architecture.

Microsoft Hyper-V

Hyper-V, or until recently Windows Server Virtualization, is a Microsoft-based product. It was originally packaged within the Windows Server 2008 bundle although, as of June 2008, it became its own standalone operating system. Hyper-V operates under individualized partitions. These partitions each execute different operating systems under a parent partition which is given direct access to the hardware resources. The guest systems run on the child partitions. The parent handles all major calls from each child and gives the child a virtualized view of available resources, limiting the actual information of the underlying hardware. The children access individualized hardware through virtual device buses, called the VMBus, which the parent then monitors. A feature present in the Hyper-V is that of the Enlightened I/O. If the guest OS supports this feature then it can bypass the VMBus for a more efficient usage of processing power as each call does not need to be emulated as in the former scenario.

Microsoft tends to take a different approach from the previous Hypervisor distributions when understanding the security of Hyper-V. The previous security documentation for Xen and VMware begin with a head-first approach to configuring specific settings to harden the system. Hyper-V, on the other hand, begins its security documentation with an analysis of the attack surface. Microsoft considers the Hyper-V architecture within its overall Windows Server 2008 package and, as such, claims that adding the Hyper-V can change the possible attack surface. It is imperative to understand this to then know where to look when thinking about Hypervisor security. They list three main areas; those being installed files, installed services, and firewall rules.

When Hyper-V is installed the original operating system that was running becomes the management OS and handles all requests for new guest instances. The actual Hypervisor utilizes a microkernelized approach to keep its footprint extremely small and disallowing any third-party code to execute on it. Microsoft states that they understand the importance of security within their Hypervisor and have "carefully reviewed and tested the Hyper-V source code to minimize this risk." That said, the rest of the document begins to detail how to configure and secure the management operating system. This is broken into two categories:

- Management Operating System Security: configuration of the physical computer itself, including discrete network interfaces for accessing the management operating system and virtual machines.
- Virtual Machine Security: configuration of the virtual machines.

The former essentially focuses on the security of the physical devices while the latter is inherently focused on the virtual devices.

One of the first recommendations to uphold when dealing with the management operating system is to set it up on a separate physical network adapter. This has been seen before in the VMware architecture as well as it provides a separate network interface solely for the management unit. This device can be further scrutinized by typical security measures, such as firewall rules and intrusion detection mechanisms, to ensure reliable network communications to the machine. They further the networking configurations by recommending front-end and back-end network adapters; the former to face the public internet and the latter facing the private intranet. Again, this is to further segregate the network to apply a more fine-grained setup of security policies for inter, and intra, network communication.

Next they move into the realm of virtual hard disks (VHDs). These files are essentially the virtual hard drives for the guest instances, which can be dynamic or static in size. When an instance is created a complimentary VHD is created alongside it. These VHD files are typically stored in a specified directory although are not required to reside there. If one were to move these files into a different directory then much caution needs to be placed into the permissions of the directory. Since the VHDs contain all data of their respective instance they need to be protected as such. Microsoft recommends a principle of least privilege when handling security permissions for these files as only administrators should have full access rights. They move on to abstract the concept of security permissions to more than just the VHDs claiming that the other files related to the instances should be protected as well. They also point out that it is good practice to segregate these files into different directories, even if they all belong to the same virtual operating system. These other files could include .ISO files, virtual machines, or virtual tape drives.

With the hardening of the management operating system taken care of one now needs to ensure security policies for the individual virtual machines. Microsoft does not give any set guidelines or configuration recommendations in this step, contrary to the previous, but instead list a set of recommendations for the administrator to think about. Some of these have already been mentioned in the previous sections, such as determining where to store the virtual machine files and the VHDs, while others have been noted in the Xen and ESX(i) Hypervisors, such as determining the amount of memory to allocate to a given virtual machine. One that has not been seen is the concept of limiting the usage of the processor. Under Hyper-V one can limit the workload any given virtual instance is allowed to do. This ensures that no given virtual machine is creating a denial of service to the processor. Another feature that is available to Hyper-V is that of time synchronization. This can be important for many reasons and should be enabled to allow the management operating system to succinctly control all given virtual instances.

To summarize, Microsoft's Hyper-V hypervisor offers a unique set of tools for managing security. They focus on two core parts of the system when dealing with hardening of their operating system. The first defines the security of the physical machine and its parts, including adapters and interfacing devices, while the second looks at policies designed to strengthen the core of the virtual machines. They claim that the Hyper-V system runs as a microkernel designed to leave a minimal footprint with the added security bonus of being unable to execute any third-party code. This grants major benefits when understanding what attack vectors are possible throughout the system.

Conclusions

To summarize, from a security standpoint each distribution has a key point of administration. For the Xen Hypervisor it is dom0, for VMware ESX(i) it is called the service console, and under Hyper-V it is known as the management operating system. These access points are the core focus in understanding the security attributes and vulnerabilities present within each distribution. The ability to monitor and understand the information flows into and out of these elevated privileged areas are paramount when determining security levels.

Each hypervisor has a different set of application-specific guidelines that they recommend although many set the same overarching policies. One that is seen throughout each distribution is that of securing the physical components of the machine and running virtual, or guest, instances in a least privileged state. Other concepts that are popular include the network installation or VLAN design and designated firewall policies. Each distribution has an individualized concept of security and each goes about it in a different light. Microsoft's Hyper-V documentation seems to take the best approach by first ensuring the administrator understand the attack surfaces, although the actual work into those descriptions seems lacking. Xen and VMware do not discuss attack surfaces although their documentation seems much more fluid and readable for system administrators to understand.

In the end each hypervisor seems to have a strong grasp on the severity of security within their systems. Each of them goes about it in different, yet similar, fashion. One glaring issue that seems to arise from each distribution, either through lack of documentation or explicitly stated, is the exploitability of virtual drivers. Xen openly explains that under certain circumstances these virtual drivers can cause known security vulnerabilities to arise while others seem to side-step the issue all together. Further work into the security of these systems could prove invaluable to the strengthening of their architectures.

Acknowledgements

First, I would like to thank Professor Judy Qui for the opportunity and motivation to produce this work. I would also like to thank the Security Informatics staff for introducing many of the topics needed in understanding the major security concepts for this work. Finally, I would like to thank the School of Informatics and Computing (SoIC) for, without them, I would not be granted these opportunities to study in this field.

Bibliography

[1] A. Tanenbaum, M. Van Steen, *Distributed Systems: Principles and Paradigms (2nd ed.)*. Upper Saddle River, NJ, USA: Prentice Hall, 2007.

[2] Xen, *Xen: Users' Manual (v3.3)*. University of Cambridge, UK, 2008.

[3] VMware Inc., *Security Hardening: VMware Infrastructure 3 (VMware ESX 3.5 and VirtualCenter 2.5)*. VMware Inc., 2008.

[4] Microsoft Corporation, *Hyper-V Security Guide*. Microsoft Corporation, USA, 2009.

[5] Xen, "Xen Hypervisor – Leading Open Source Hypervisor for Servers," http://www.xen.org/products/xenhyp.html

[6] Intel, "Intel Virtualization Technology," http://www.intel.com/technology/itj/2006/v10i3/1-hardware/6-vt-x-vt-i-solutions.htm

[7] AMD, "AMD Virtualization (AMD-V) Technology," http://sites.amd.com/us/business/it-solutions/virtualization/Pages/amd-v.aspx

[8] VMware Inc., "VMware ESXi and ESX Info Center," http://www.vmware.com/products/vsphere/esxi-and-esx/index.html

[9] VMware Inc., "Benefits of VMware ESXi Hypervisor Architecture," http://www.vmware.com/products/vsphere/esxi-and-esx/why-esxi.html

[10] VMware Inc., “VMware ESX and ESXi 4.1 Comparison,” http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1023990