Security Benefits from OS Virtualization: Real or Virtual?

Apostol T. Vassilev[1]

Abstract. **Recently, people have begun to use OS virtualization as a tool for improving LAN security. While virtualization is very useful in optimizing hardware utilization, we show that its security benefits come at a price.**

I. Introduction

Operating system (OS) virtualization allows businesses and individuals alike to use their available computer hardware resources much more efficiently and flexibly. This technology has become indispensable for many professional software developers and testers, allowing quick configuration of reference OS images that can be used in different contexts, often executing on the same computer [1].

There are two main technological approaches to OS virtualization [2]: standard and lightweight; lightweight further splits into *containers* and *paravirtualization*. Each approach has different architectural and run-time characteristics, hence different robustness of the isolation from the host OS.

Recently, an important security trend has emerged in OS virtualization usage. Consumers have begun to use virtualization as a tool for isolating processes, e.g., Internet browsers, in order to prevent malicious software (malware) from invading their main computing environment [2]. Using any of the available OS virtualization technologies, one can configure an image of an operating system with a browser and execute it on a computer as an isolated process within the host OS. A user can then utilize the browser in the image to explore the Internet without a fear that the main system will be invaded by malware. But are users safe, really?

In this paper we consider the problem of process/application isolation through OS virtualization and

[1] A.T. Vassilev is with NetIDSys, Inc., 3405 Oxen Court, Austin, TX 78732. E-mail: apostol@netidsys.com

examine how it can be used in practice for securing users' computing environments.

II. OS VIRTUALIZATION APPROACHES

OS virtualization technologies tend to differ from one another based on the comprehensiveness of the *hypervisor* module. The hypervisor emulates the hardware devices and manages the system resources allotted to each VM. It can run as an application or be built in the host OS (see Figure 1).

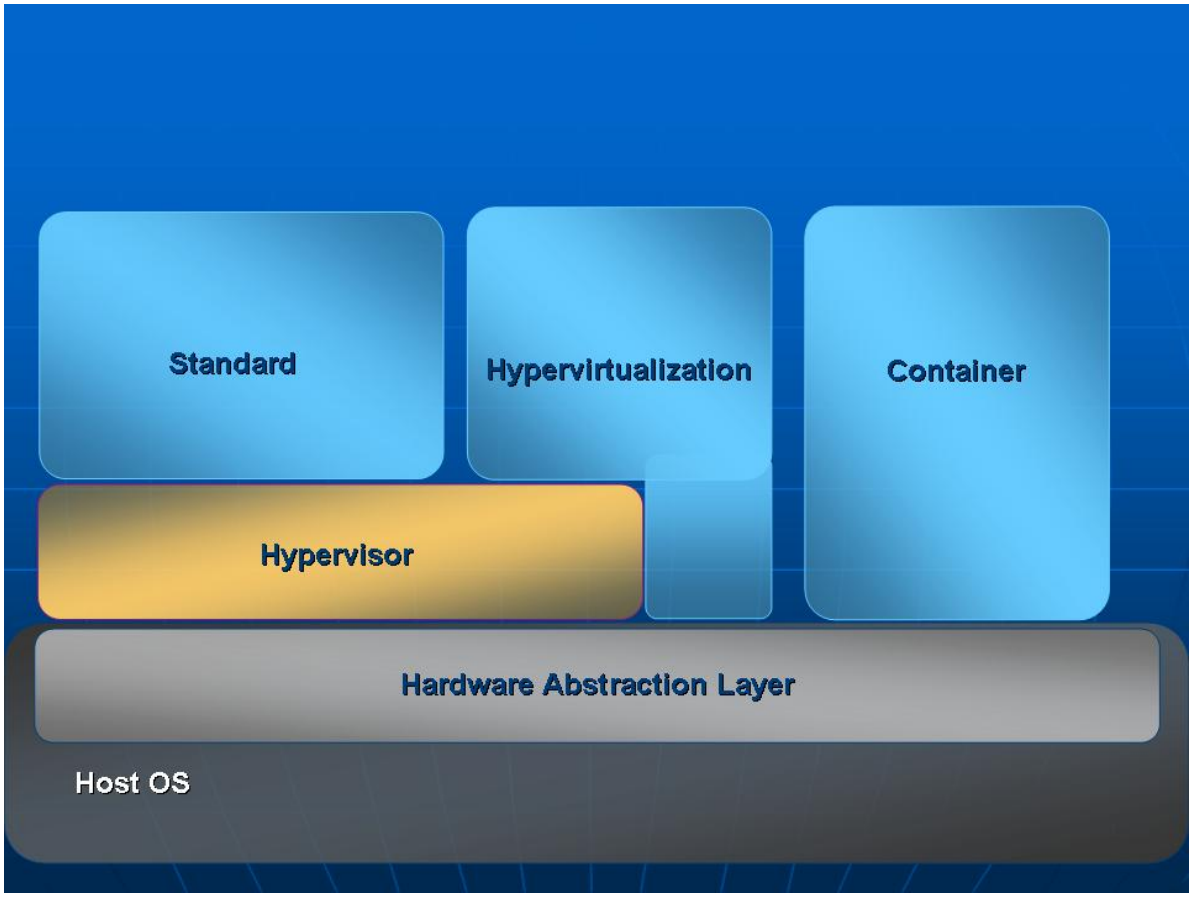

Figure 1. OS virtualization approaches

A. Standard virtualization approaches

Virtualization approaches with comprehensive hypervisor implementations, capable of emulating all hardware devices, are often referred to as standard virtualizations. Examples of commercial systems for

standard virtualization are VMware®[2] for Windows and Linux, and Microsoft's Virtual PC (VPC) for Windows®[3]. The VPC technology does not allow direct access to the host OS hardware but the list of supported devices is less comprehensive (USB devices or smart card reader devices are not included). However, when supplemented with Microsoft Windows Remote Desktop Connection (RDC), the resulting combination has extended device emulation capability: RDC allows the user to configure additional local devices for emulation to the remote Windows instance it is connected to. So, if the local system is the host OS and the remote desktop is a Virtual PC image executing within, the resulting combination is a standard virtualization. Note that RDC can be used with any of the other Windows OS virtualization technologies, e.g., VMware.

B. Lightweight virtualization approaches

Virtualization approaches with less than complete hardware emulation in their hypervisor implementations are called *lightweight*. This technology tends to offer more efficient use of the available hardware under the assumption that each VM is serving a purpose of limited scope and can be constrained within limited system resources and a small set of emulated devices needed for the particular task.

The *container* approach is built in response to the need for increased isolation between groups of processes running inside the host OS kernel. A container virtualizes the host operating environment and isolates processes clustered within but does not emulate any devices. Any application running inside a container calls directly into the underlying host OS to access the hardware resources. In essence, the applications in one cluster are isolated from the processes in another cluster within the operating environment of the kernel OS, but they are not necessarily isolated from each other at the hardware resource level, i.e. shallow isolation. This observation is important for our security considerations. The container approach is best suited for running multiple software applications within their native host OS.

The *paravirtualization* approach not only virtualizes the operating environment but also provides selective emulation of hardware devices for the applications inside a cluster. Intuitively speaking, this

[2] VMware is a trademark of VMware, Inc., an EMC company.

isolation is potentially deeper than the container isolation and therefore more interesting from a security perspective. Most commercial paravirtualization systems allow flexibility in configuring applications' access to hardware resources, ranging from full hardware emulation, and therefore complete isolation from the host OS, to direct access to the hardware resources of the host OS. Paravirtualization is equivalent to cluster virtualization on the latter end of the configuration spectrum, whereas it resembles standard virtualization on the former end.

III. SECURITY BENEFITS OF VIRTUALIZATION APPROACHES

As we saw in the overview above, the emulation of the available host OS hardware resource is the main criterion that differentiates the OS virtualization technologies. From a security perspective, solutions that emulate and thus isolate the underlying host hardware devices tend to be more secure than those that afford direct access. For example, a virtualization technique that emulates a hard disk as a data file contained within the real file system of the host OS provides a level of isolation that can be very useful in deterring attacks. In this case the damage inflicted by many viruses and malicious tools will be contained inside the data file and so it will be harmless to the applications inside the host OS that generally do not operate on the infected file.

This is one of the main reasons for the growing popularity of virtualization in defending against the onslaught of Internet attacks. Many businesses and consumers are turning to virtualization as a safe environment for exploring the Web. People expect that running browsers and instant messaging (IM) tools inside a VM provides a level of insulation capable of containing the majority of malicious exploits coming from the Net. The user community at large has begun to view the virtualization technology as an effective tool against day-zero attacks, in particular with regard to the emerging threat coming from a new wave of obfuscated computer viruses. This kind of malware renders useless the existing anti-virus technology that relies on fixed code signatures for detection. In contrast, obfuscated malware utilizes cryptographic techniques to masquerade a random code signature, thus defeating the standard virus

[3] Windows is a trademark of Microsoft Corporation. VPC and RDC are Windows components.

protection.

To continue the investigation of the security benefits of OS virtualization we need to define a threat model.

A. Threat models for OS virtualization

Formal security analysis [6] of software systems is the most solid approach to security evaluation of operating systems and applications. It requires access to detailed design documentation and source code based on which a formal model is derived. In most practical cases of interest the derivation and analysis of the corresponding formal models is extremely difficult due to the complexity of the modeled systems, even if OS and virtualization tool vendors were willing to provide both documentation and source code.

In the absence of detailed information about the system one has to resort to an empirical technique for security analysis known in the trade as *threat modeling*. It consists of compiling a set of possible attacks against the system that a hacker may undertake and corresponding counter measures. Often, it is useful to decompose a complex system into components and then define threat models for each component as well as models for the interactions of the different components.

Intuitively speaking, the more complete the threat model, the harder the system is to hack, provided it implements effective counter measures for each of the listed attacks. In reality, formal definitions of completeness for a threat model, from which practical guidelines for compiling the list of attacks can be derived, are difficult to come by. Consequently, the completeness of a given threat model is judged subjectively by the individuals or organizations evaluating a particular system.

Our goal in this paper is to evaluate the protective capabilities of the VM used as a shield against threats to the host operating system. We consider a canonical two-component system (C1 - the host OS; C2 - the VM executing within C1) with three interfaces: I1 - the host OS interface to the local area network (LAN); I2 - the virtual machine LAN interface; I3 - the interface between the virtual and host OS instances (see Figure 2).

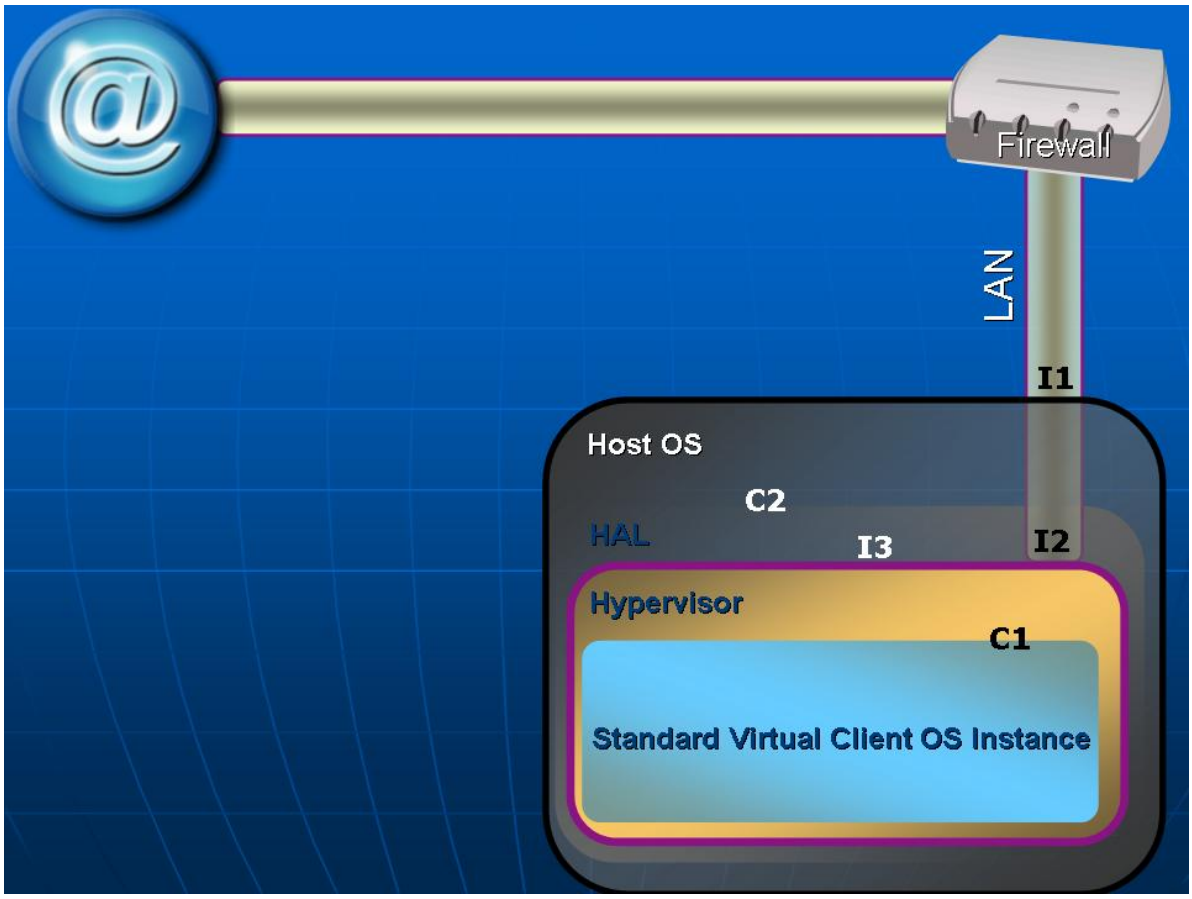

Figure 2. Canonical system: two components (C1, C2) and three interfaces (I1, I2, I3)

We consider attacks that originate from the Internet in the form of malware or viruses that spread through browsing activities, e-mail, etc. In particular, we are interested in attacks that come through the I2 interface. Attacks that infect the VM but fail to penetrate into the host OS (or the LAN) through I1 or I3 we consider as failed or contained.

Recently, the possibility for instigating theoretical attacks based on inserting a malicious hypervisor into the host OS image has been considered [3], [4]. Although interesting in their own right, such attacks are not in the scope of this paper.

There are two types of malware we consider: type I - malware incapable of detecting the exact characteristics of the invaded operating environment and not acting to invade the LAN through I2 or penetrate the hypervisor through I3; type II - malware capable of detecting the operating environment

characteristics and acting to penetrate its boundaries. Most known malicious software tools and viruses are type I. Many network-savvy viruses designed to infiltrate the LAN are type II.

To date there is no known malware specifically designed for virtual OS instances even though it is relatively easy to detect that a particular operating environment is a VM and act accordingly. With the popularity of OS virtualization growing rapidly, it is prudent to take into account such threats as well. Note that here we refer to viruses penetrating a configuration consisting of a legitimate hypervisor, VM, and a host OS, not the malware considered in [3] and [4].

B. Security evaluation of virtualization approaches

The container approach in lightweight virtualization affords direct access to hardware resources to applications executing within. Therefore, a virus invading a particular container has access to the same underlying potentially-sensitive hardware resources as the host OS: hard disks with user and system data. This virtualization approach is best suited for hardware utilization optimization and should not be used as a security process isolation tool equivalent to standard virtualization.

Similarly, for hypervirtualization systems to provide valid secure process isolation, they must be configured for emulation of all supported hardware devices. In other words, processes executing within the virtual OS image must not have direct access to host OS resources. At the same time, people should be very careful about the types of devices they configure for emulation. For example, the RDC configuration for Windows platforms mentioned earlier allows remoting of host OS disks and smart card devices to the virtual desktop. This is a potential security hole because viruses of type I and II may get direct access to valuable resources in the host OS. RDC should be configured without emulation of local hard disks when used for security isolation purposes.

Standard OS virtualization offers the best computational environment for configuring secure process isolation. We consider two possible network configurations with different protective potential.

1) VM clients on the LAN

Here we consider the canonical configuration from Section III.A (cf., Figure 2) with the VM setup as a

client system with access to the Internet through a LAN firewall. The typical VM configuration people employ is the default network configuration established by the VM vendor, which may be a potential gate for malicious attacks. It is reasonable to expect that such a VM will be able to deter type I malware but it must be tightened to close all potentially-vulnerable network ports in order to resist type II malware. Each OS vendor offers network port configuration tools with matching security guidelines and the VM must be configured accordingly. If a client VM on the LAN is compromised by type II malware, it may spread the worms just like other regular clients connected to the same network. To protect against such a problem, organizations may utilize node-based firewalls that block potentially vulnerable ports on all LAN nodes. Alternatively, people may use IPSec security policies to block problematic ports but to be effective a policy must block all vulnerable ports completely or, at least, require security over such ports with hosts that are completely trusted. Certainly, a shielding VM should not be trusted on the LAN. Virtualization technology vendors have also begun to offer pre-configured images that are tightened with respect to network port availability.

Unfortunately, although useful, strengthened images do not provide complete isolation. The degree of protection largely depends on the discipline of policing the image once deployed within the LAN. The problem here is that the isolated image is rarely useful on its own. Users often want to transfer content from the VM to other computers on the LAN. In practice, virtual machines may have several mounted network disks. In fact, any shared disk drive on a computer connected to the LAN is a potential target. While this problem may be mitigated by enforcing strict security policies on the corporate LAN, the severity of this potential weakness is almost unlimited in the home LAN, where most people lack the expertise and/or discipline to eliminate it. The canonical network configuration in Figure 2 is not likely to hold up against day-zero type II exploits. However, home and corporate users can benefit from VM technology with even default configuration to protect themselves against type I malware as well as the IM information ex-filtration problem [5], assuming the IM tool is not compromised. Many businesses in possession of sensitive data are not allowing IM tools on the corporate networks because, at best, such

tools make it very difficult to separate malicious information ex-filtration from innocent IM traffic; at worst, such tools are often back doors for malware infiltration. These potential problems are made even harder when the IM tool communicates over encrypted channels.

Indeed, the IM tools have access to only local hard disks and the network interface I2. In the case of a well-isolated VM the local disks are only virtual disks. Also, if the IM tools are legitimate, they do not try to abuse the other client machines on the LAN and use the I2 interface only to reach the remote server of the corresponding service provider. So, any information potentially leaked outside of the VM will be data contained within, i.e. not too valuable, the communication logs that the tool itself owns notwithstanding. Of course, if an IM tool turns into a back door for type II malware infiltration, the VM isolation in this network topology may not be sufficient to prevent the viruses from spreading in the LAN. A different network topology capable of resisting such attacks is considered in the next section.

2) VM servers outside the LAN

Perhaps the cleanest and most secure configuration for protecting against types I and II malware is a network with a node, typically a server with one or more virtual machines on it, outside of the LAN firewall (see Figure 3). Users connect to a virtual server instance hosted on the dedicated server using some kind of a thin client utility, e.g. Citrix®[4] or RDC.

[4] Citrix thin client is a registered trade mark of Citrix Systems, Inc.

Figure 3. Network topology with an isolated server VM

This configuration is quite useful as an Internet security shield and not too difficult to set up for most enterprises. In fact, if one VM is compromised, the worms will be filtered by the LAN firewall because the traffic from it will be treated as external Internet traffic and subjected to stringent quarantine rules. The occurrence of such incidents or any other network traffic irregularities within the hosting server should be monitored closely, in particular to detect day-zero types I and II exploits. Strict security policies must be enforced to avoid known traps: e.g., users should not be allowed to remote local hard drives to the VM outside the firewall. The security policies should also prevent client machines from bridging networks in different security layers, by being connected to two such networks at the same time, for example. Note also that corporations can use this configuration to eliminate the IM information ex-filtration risks [5], while still allowing their users to benefit from IM services.

In reality, the network topology shown in Figure 3 is not exclusive for VM technology. It can be

implemented with a regular server hosting remote login for thin clients with its native OS. In this case the main benefit from OS virtualization on the server is the possibility for optimization of hardware utilization, e.g., by running several different hosting services on the same hardware.

Unfortunately, the security benefits of the network topology shown in Figure 3 are of little consolation to the home user because it involves components that are prohibitively complex and expensive for most people.

When it comes to detecting the operating environment of virtual OS images, this problem is not hard. For example, in the case of VMware, the running image's network configuration defaults to well-known IP addresses. So, it is possible to think that with the popularity of virtualization rising, it is only a matter of time before viruses capable of attacking the I3 interface appear. This means that the porosity of the hypervisor will be put to test. To be useful as a shield, the hypervisor should provide comprehensive emulation of many different devices supported by the host OS. However, the more complex the component, the more likely it is to find vulnerabilities within, due to design or implementation deficiencies.

IV. Conclusions

OS virtualization technologies are very useful in optimizing hardware utilization but the benefits they bring for improving the security of the digital information assets on the LAN come at a price. For most practical applications, this requires: first, choosing the right virtualization technology, with the standard approach being most appropriate for security purposes; second, carefully planning, configuring, and enforcing stringent LAN security policies to achieve the security improvement goals a corporation or an individual may have. The complexity of the configuration process and the need for stringent policy enforcement make such solutions more suitable for enterprise applications.

There are some cases where VM technology can easily improve the security and privacy protection, namely defense against type I malware and the information ex-filtration problem [5], but in general people should be careful not to place too high security hopes on this technology.

REFERENCES

[1] D. Fisher, "*Virtual threats*", Information Security, pp. 46-51, January, 2007.
[2] S.J. Vaughan-Nichols, "*New Approach to Virtualization is Lightweight*," IEEE Computer, pp.12-14, November, 2006.
[3] J. Rutkowska, "*Subverting Vista kernel for fun and profit*," Proceedings of BlackHat USA Conference, July 29-August 3, 2006.
[4] S. King at al., "*SubVirt: Implementing malware with virtual machines*," Proceedings of IEEE Symposium on Security and Privacy (the Oakland conference), May 2006.
[5] P. Biondi and F. Desclaux, "*Silver Needle in the Skype*," Proceedings of BlackHat Europe Conference, March 2-3, 2006.
[6] A. Menezes, P. van Oorschot, and S. Vanstone, "Handbook of Applied Cryptography," CRC Press, 1997.