Virtual Security
About the Security Pros and Cons of Server Virtualization

Martin Wimmer

Siemens AG, Corporate Technology, CT IC CERT
D-80200 Munich, Germany
martin.r.wimmer@siemens.com

Abstract. Recently, the discussion about security of virtualized IT infrastructures has intensified. Several research papers have been published discussing both, the pros and cons of virtualization for security. Additionally, new business ideas and products have been developed for enhancing security for virtualized IT. With this paper we provide a survey of the recent advances in computer security for server virtualization.

1 Introduction

In the recent years, virtualization – known since the 1960s – has experienced a revival. Virtualization solutions are increasingly used in data centers and for desktop applications, aiming at lower total costs of ownership and flexible ways of hardware administration. Additionally, virtualization can also improve system stability and security. However, if not configured and applied thoroughly, significant security risks may ensue.

A brief overview over virtualization techniques will be given in Section 2. Virtual machines (VMs) decouple the system running in the VM from the underlying host, both regarding the hardware and the host's configuration as well as the host's reliability and stability. This kind of decoupling can be used for security purposes, e.g., for improving disaster recovery capabilities, providing high availability, or sandboxing malicious activities as will be discussed in Section 3. On the other hand, virtualization – when configured imprudently – can have negative impact on the overall system stability and availability as discussed in Section 4. What is more, virtualization software introduces an additional architectural layer that can be target of attacks. Section 5 highlights attack scenarios that range from VM detection (e.g., to hinder malware analysis) and denial of service on the running VM up to circumventing the VM's isolation mechanism. In the worst case attackers may gain control over the host platform and, possibly, other virtual machines running on the same host.

Often, the security requirements and risks of virtualized IT infrastructures are insufficiently taken into account – assuming that virtualized systems are secure by default. We will therefore identify basic guidelines for securing virtualized computing environments, for example, covering adequate security policies for the use of virtualization in a corporate environment as well as more technical issues such as network configurations (see Section 6). Section 7 discusses novel

approaches for providing security for virtualized environments and Section 8 concludes the paper.

2 Desktop and Server Virtualization Techniques

Virtualization denotes the creation and management of so-called *virtual machines* (VMs) through combinations of hardware and software. Virtual machines represent abstractions of computer resources such as a server, an operating system, an application, or a storage device. Through virtualization, single physical resources can, for example, appear to function as multiple logical resources or as resources providing different characteristics/features (so-called emulation). In this regard, several VMs can co-exist on the same physical resources whereby – following the pure principles of virtualization – VMs are isolated from each other and from the host systems they are executed on.

Server virtualization denotes the virtualization of complete computing platforms, i.e., hosts. That is, a VM realizes an abstraction from the underlying physical host, providing mediated access to both virtual and real hardware. Guest operating systems can be installed on VMs, acting as if they were running on real physical machines. In theory, the environment provided by the VM is self-contained, isolated, and indistinguishable from the hosting machine. The creation of VMs can, for instance, be based on emulation, full virtualization, or paravirtualization.

2.1 Emulation

The virtualization layer emulates, i.e., simulates, the complete hardware including CPU, chipsets, I/O-components, etc. This allows an unmodified guest operating system to even run on (simulated) hardware which is different to the actual physical hardware. Examples include QEMU [T01], Microsoft Virtual PC for MAC [T02], the Hercules emulator [T03], and Bochs [T04]. Fields of application for emulators are, for instance, development processes, in particular those for implementing and testing software for hardware configurations different to the development system.

2.2 Full Virtualization

The virtualization layer – called *virtual machine monitor* in this context – simulates the hardware which is necessary for unmodified guest operating systems to run in isolation. That is, the CPU is passed through to the VMs while other hardware components like memory, disks, network adapters and other I/O devices are emulated. Typically, more than one VM instance can run simultaneously on a primary OS with their number being limited only by the host computer's hardware and memory resources that are to be shared. In many scenarios, virtual machine emulators are used to run operating systems different to the one of the host computer. Well known representatives of such virtual machine emulators

are Microsoft Virtual Server [T05], VMware Workstation, VMware Server [T06], and QEMU [T01].

2.3 Paravirtualization

Using paravirtualization, virtual machine emulators – usually called hypervisors – not necessarily simulate hardware, but instead (or in addition) offer a special API for access to the hardware. This implies, that the guest operating systems need to be modified. Such system/hypervisor calls are referred to as hypercalls in Xen [T07]. Examples of distributions providing paravirtualization are Fedora Core Linux installations with integrated Xen hypervisors and VMware ESX Server [T06]. Compared to full virtualization, paravirtualization offers improved performance for guest operating systems.

2.4 Hardware-supported Virtualization

In addition to pure software-based emulation, virtualization capabilities can be provided by the hardware layer. Examples for the hardware-supported virtualization of x86-based operating systems are the Intel Virtualization Technology, Intel VT for short (codename 'Vanderpool'). The 32-bit Intel VT extensions are referred to as VT-x and the extensions of the 64-bit Itanium processors also named VT-i. AMD also provides hardware-based virtualization by means of the AMD Virtualization extensions, abbreviated as AMD-V and known under the codename 'Pacifica'. The use of hardware-based virtualization is often supported by full virtualization and paravirtualization software.

3 Security Benefits of Virtualization

Virtualization offers a lot of operational benefits, like improved hardware usage and flexible server administration concepts. In the following we focus on the benefits of virtualization from the point of view of security.

3.1 Security by Virtualization

The stability and overall security of IT infrastructures can benefit through the use of server virtualization. The most significant key factors this relationship is based upon are the isolation and snapshot capabilities of VMs.

– **Isolation of unstable or compromised applications** A basic characteristic of virtualization is that the individual virtual machines are running in isolation, i.e., within a confined environment. That is, the stability/instability of one VM by design does not have an impact on the stability of other VMs which are executed on the same physical host. The concept of isolation provides security for the following scenarios:

  - Software which is considered to be instable or even untrustworthy can be executed in isolated environments without causing potential harm for production systems.
  - The mentioned sandboxing functionality allows confining compromised systems, hence, hindering the distribution of malware.
  - The isolation capability is also useful for the evaluation and testing of new product versions and updates, e.g., during software development processes or patch cycles.
- **Separation of security functionality from production systems** Malware often tries to disable security functionality in order to conceal itself from the user, thus, being able to perform malicious actions behind the scenes. An example might be the deactivation of firewall services which otherwise would alarm in case unusual connections to untrustworthy endpoints are established. Using separate VMs for production servers and intrusion detection/prevention systems, security functionality cannot be deactivated in case production systems get compromised.
- **High availability** In corporate environments, the preservation of data and the availability of services are essential. Virtualization can contribute to reducing the time and costs for disaster recovery through using VM instances for failover purposes. In this regard, as the failover services are usually idle in normal phases, several hot standby installations can be hosted on the same servers – at least under the assumption that systems are unattached so that the outage of one system will not crash other systems whose failover pendants are all running on the same physical machine.
- **Disaster recovery** Virtualization allows to remedy system compromise through supporting the fast and easy reset to clean and trustworthy installations. When you have to deal with zero-day exploits, rootkits and advanced spyware and adware programs, usually the only reliable solution is to throw away the infected system and restart from scratch. With regard to this, server virtualization offers significant advantages. What is needed are (trustworthy, i.e., non-infected) baseline installations of the affected systems. After evaluating the compromised system, the baseline installation can be patched and turned into production.

3.2 Virtualization-based Security Applications

Virtualization can also be employed to realize specific security projects.

- **Malicious code research** VMs can be very helpful in analyzing malicious code samples, retrieving information about their impact [1]. The idea behind using virtualization for malicious code research is that if the virtual machine gets infected with malware or infiltrated by an attacker, it is not possible to escape to the host operating system. Erasing an infected VM and reloading it from a trusted image is a rather simple and fast way of resetting the test environments. In doing so, the risk of infecting the underlying host operating system is quite low, at least if both, the host as well as the guest system are hardened.

- **Forensic analysis** Virtualization also eases computer forensics. Clones of potentially compromised hosts can be created and afterwards run in VMs and analyzed without the need to provide a computing architecture that is similar to the analyzed host.
- **Realization of honeypots and honeynets** Individuals and organizations that run honeypots or honeynets are also attracted to server virtualization [1]. Honeypots are services which are used to monitor attack attempts. Usually they provide enough interesting data to attract attackers but are not used by standard users of the network and are equipped with lots of logging sensors. Hence, the fact that someone tries to interact with a honeypot is regarded as a potential attack.
- **Creation of intrusion detection tools** Virtualization also offers a cheap and easy way of setting up intrusion detection and prevention tools, respectively. Instead of purchasing dedicated hardware for these purposes, preconfigured software installations can be downloaded (often without fee) and executed in VMs.
- **Testing of new patch releases** Systems need to be patched in order to close known vulnerabilities. Nevertheless, the usability and stability of systems might suffer in case patches are not stable or reveal potential side effects. Therefore, in many cases patches are not applied nowadays, exposing systems to avoidable security risks. In order to avoid these drawbacks, patches first can be tested in a VM before being installed on production servers.

4 Security Limitations of Virtualization

This section gives an overview of recently reported security challenges for virtualized IT infrastructures.

4.1 Service Drop-outs in the Large

Virtualization represents a helpful technique for consolidating different systems onto the same physical servers, resulting in a reduced number of hosts that have to be purchased and maintained. Nevertheless, from the point of view of security, applications running inside a VM depend on several layers of the underlying infrastructure. These layers, for example, consist of the physical machine, the hypervisor or virtual machine emulator – running on a host operating system –, and, of course, the guest operating system. In contrast, considering traditional IT infrastructures, there are only dependencies from the physical machine and the operating system. Thus, up to two additional architectural layers exist which can cause system instability.

While in a traditional computing environment services are often exclusively assigned to servers, i.e., one physical host for one dedicated service, virtualization is oftentimes used to package several services onto the same machines. However, the consequences are obvious: If no additional measures are taken,

several services will suffer from drop outs in cases when the physical server, the host operating system, or the virtual machine emulator fail. Therefore, virtualized environments – more than traditional systems – demand for a profound patch management, continuously closing newly detected vulnerabilities.

4.2 Uncontrolled Growth

As setting up VMs is rather easy and virtual systems can easily be copied and shared, the number of VMs can increase drastically in a corporation and be a multiple of the number of physical machines. Apart from the mere number of systems which can arise and disappear again, security officers are confronted with a potentially high diversity of operating systems. Virtualization, respectively emulation, allows running different types of operating systems or systems of varying versions and patch levels. What is more, VMs are often used for testing purposes and are managed by non-administrators. Thus, a highly heterogeneous landscape might be given including lots of unpatched, unsupported and insufficiently administered and secured systems. Thus, server virtualization literally can result in "robbing Peter to pay Paul": While the physical infrastructure can be homogenized, the operating system diversity can reach an unmanageable growth. This creates a range of problems as one must try and maintain patches or other protection measures for a diversity of operating systems or deal with the risks posed by tolerating unpatched machines on the network.

In traditional computing environments, machines are usually uniquely identifiable and associated to their (main) users or administrators. The same does not necessarily apply to virtual machines. What is more, using bridged network configurations, VMs can be integrated into networks with their own network identities, i.e., MAC and IP addresses. Such preconditions make it more difficult to keep an up-to-date view on the system landscape.

4.3 Impact of Snapshot Functionality

In virtualized environments system snapshots can rather easily be made. This allows to secure stable system states and to easily perform rollbacks. However, such useful mechanisms can also cause unforeseen but harmful side effects.

- In traditional computing environments, systems are usually iteratively patched, updated, and configured. Hence, a system's lifetime can be compared to a straight line. In contrast, in a virtual computing environment, different instances (representing different versions) of the same system with different patch levels can coexist [2]. Providing a comparable level of security for virtualized IT infrastructures will thus result in significant management overhead to ensure that all instances are up-to-date.
- Through virtualization features like cloning and snapshot functionality, known security vulnerabilities that have already been fixed, can be reintroduced. That is, rolling back to a system state before patches have been applied, it is possible to reintroduce malware like worms, viruses, and Trojan horses, lose new firewall or HIPS rule sets, or reactivate vulnerable services.

– Due to the snapshot functionality complete systems like web servers or database servers containing confidential data are available in the form of a few files. These can be copied – e.g., on large mobile storage devices like USB disks – and distributed easily. The systems can then be restarted / modified / analyzed on any other host providing an appropriate virtualization platform. Hence, mobility eases the theft of confidential data as complete systems can be stolen and analyzed offside.
  Another aspect is that security relevant information might remain longer in a system or is stored persistently without the intent or knowledge of the users: In order to implement rollback functionality, the virtual machine monitor has to log the system state. Thus, data that is supposed to be volatile or deleted might remain in the system, or even stored persistently due to memory management (e.g., swap files) or the creation of snapshots, hence, posing a risk to data confidentiality.

5 Attacks on the Virtualization Layer

A virtual machine emulator represents an additional architectural layer which can suffer from security vulnerabilities and be the target of attacks. For example, in September 2007, it was reported that VMware released a cumulative security patch,[1] showing that VMware products were vulnerable with regard to several security threats. At the end of 2007, Lamb published an overview of vulnerabilities applying to VMware (accumulate for all VMware products) which is shown in Table 1. The vulnerabilities were categorized by severity, by impact, by vector and by whether the vulnerabilities were in VMware's proprietary first-party components or in third-party components that are used in VMware products. This table illustrates that of the 100 vulnerabilities in total, 57 were remotely accessible and 46 represent high risk vulnerabilities. That already represents a strong motivation for having a patch management for virtualization platforms in place.

Of course, security vulnerabilities are not unique for VMware products only and other virtualization products suffer from potential security leaks and attacks, too. Following Peter Ferrie [3], attacks can be categorized into: (1) detection of VM presence to conceal malicious code activities, (2) Denial of service on the virtual machine emulator, and (3) – which is considered to be the most challenging and threatening type of attack – VM escape.

5.1 Detecting VM Presence

As illustrated in Section 3.2, VMs are willingly used to analyze malicious software. However, if malicious code detects that it is executed within a VM, it can hide its impact by changing its behavior or refusing to run. Hence, using virtualization to analyze malware is possibly misleading or at least more complicated. As shown by several recent research papers, various ways exist for detecting the presence of VMs:

[1] `http://archives.neohapsis.com/archives/fulldisclosure/2007-09/0356.html`

VMware Vulns. by Year	Total Vulns	High Risk Vulns	Remote Vulns	Vulns in 1st Party Code	Vulns in 3rd Party Code
1999	1	1	0	1	0
2000	1	1	0	1	0
2001	2	0	0	2	0
2002	1	1	1	1	0
2003	9	5	5	5	4
2004	4	2	0	2	2
2005	10	5	5	4	6
2006	38	13	27	10	28
2007	34	18	19	22	12
Totals	100	46	57	48	52

Table 1. Temporal course of VMware vulnerabilities since 1999 (source: [4])

5.1.1 Detecting hardware-supported virtualization Hardware-assisted virtual machines use CPU-specific instructions to place a system into virtual mode. On Intel processors, its the VMLAUNCH instruction and on AMD CPUs the VMRUN opcode, respectively. Guest operating systems run at the same privilege level as they would do if they have full control over the CPU without any virtual machine running. Actually, the virtual machine layer (hypervisor) is more privileged than the host OS as it sees intercept, interrupt, and exception events first and can hide them from the host operating system. Instructions (including the CPUID instruction[2]) which would leak information about VM presence see only shadow copies of sensitive data structures which appear to correspond to real CPUs.

Therefore, the presence of hardware-assisted virtual machine layers are hard to detect but not impossible. The approaches used to detect hardware-assisted hypervisors usually rely on timing. They make use of the fact that executing certain instructions repeatedly (many times) takes longer within a VM than without. For example, accesses to data structures before and after triggering hypervisor events can be timed. On Intel CPUs the Translation Lookaside Buffers (TLBs) will be flushed when calling hypervisor-sensitive instructions like CPUID. Similar possibilities exist for the L2 cache on Intel and AMD CPUs. Then, the time for fetching something from memory before and after executing hypervisor-sensitive instructions can be measured and compared. If both deviate, VM presence is likely. Timing approaches demand for a comparison to executions without the presence of a hypervisor or require external time sources.

5.1.2 Detecting Software-based Virtualization Several approaches have been presented for detecting the presence of software-based virtualization like VMware or Virtual PC [5,6]:

[2] By using the CPUID opcode, software can determine the processor's type and features.

- *Determining VM artifacts in processes, the file system and/or the registry* is a rather straightforward and simple approach. Virtualization products like VMware offer several possibilities to detect whether VM instances are running. Examples include running processes (e.g., VMtools), file system entries referring to telltale files and folders, and diverse entries in the registry.
- *Determining VM artifacts in the memory* refers to VMs leaving their marks in the main memory. Considering VMware, for example, this includes identifying string values. Instead of checking the complete memory for identifying string values, a promising alternative is to examine the location of system data structures. On host machines, data structures like the Interrupt Descriptor Tables (IDT) are located at lower addresses in memory than their counterparts on guest systems. Further data structures supporting this approach are the Global Descriptor Table (GDT) and the Local Descriptor Table (LDT, [7]). Tools making use of such checks are, for example, *Red Pill* developed by Joanna Rutkowska[3] and *Scoopy Doo* [T09].
- *Looking for VM-specific hardware devices* is also a promising approach. Virtual machine emulators usually introduce VM-specific hardware, such as hard disks whose device names are constant, and network cards whose MAC addresses are within a predefined range/start with identifying prefixes.
- *Looking for VM-specific processor instructions and features* can be done in two directions. On the one hand, non-standard instructions which are not included in the instruction set of x86 architectures can be determined. On the other hand one can look out for specific guest-to-host communication channels. Following the first possibility, VMDetect [T08] registers its own handler for invalid opcodes. Afterwards, it executes VirtualPC-specific, non-standard instructions which will cause exceptions (thus invoking the handler) in case VirtualPC is not present and none otherwise. Hence, if no exception is thrown, a Virtual PC-based VM is present. With regard to the second approach, VMware uses a specific method for guest-to-host communication – an undocumented feature which can be exploited for VM detection. Please refer to `http://www.trapkit.de/research/vmm/jerry/index.html` for further information on this detection technique.

Please note that VM detection cannot be avoided reliably, though the titles of publications like [5,6] suggest the opposite. The mentioned papers mainly focus on VMware specific features and, for example, discuss possibilities for concealing the presence of VMware specific guest-to-host communication channels. Nevertheless, as VM detection based on IDT, GDT, or LDT locations in memory is quite simple, the presented approaches are little promising. Moreover, they are difficult to apply and not yet robust.

5.2 Denial of Service Attacks

Going beyond detection, virtual machine emulators can be targets of attacks with the objectives to reduce the usability of VMs. That means classical de-

[3] `http://invisiblethings.org/papers/redpill.html`

nial of service (DoS) attacks resulting in abnormal terminations of VMs or high computational load (e.g., produced through infinite loops) which hinder users or administrators to interact with affected VMs. In particular VMware has been shown to suffer from several DoS vulnerabilities.[4] In the past, few research papers, like the one provided by Peter Ferrie [3], have been presented describing possible DoS attacks on virtual machine emulators. Among others, he describes a technique to cause a fatal error on Parallels instances causing them to terminate abnormally. Further, a good overview of possible attacks on diverse virtual machine emulators is provided by Travis Ormandy [8].

5.3 Virtual Machine Escape

Virtual machine escape denotes that the virtual machine emulator is subverted to execute arbitrary code on the host system with the privileges of the virtual machine emulator. This denotes a total compromise. Compared to VM detection, VM escape is a rather difficult and challenging type of attack and only few vulnerabilities providing the basis for VM escape have been reported so far, for example:

- Several virtual machine emulators suffered from security leaks which could be exploited to escape from the protected environment of a VM (see [8,9]). Considering VMware, a security hole of the `vmnat` service was reported at the end of 2005.[5] It was shown that specially crafted EPRT or PORT FTP commands result in heap overflows providing a basis to compromise the host system.
- In September 2007, VMware released a cumulative patch for multiple denial of service vulnerabilities. That is, through flaws in VMware implementations, denial of service attacks against the host operating systems can be performed.[6]
- In Section 5.1.2 we mentioned that VMware's guest-to-host communication channel can be used to detect the presence of VMs. Though it is an undocumented feature, it has been reverse engineered successfully. Theoretically, this communication channel can be misused to steal data (e.g., data stored on the clipboard) and to query sensitive information about the host.

Fortunately, exploits and proof of concepts of attacks are rare. Nevertheless, VM escape represents a critical attack scenario offering attackers extensive control possibilities. Akin to popular operating systems, virtual machine emulators suffer from similar problems of a usual software lifecycle so that further vulnerabilities and exploits are likely to occur in the future.

[4] `http://www.securityfocus.com/bid/23732/info`
[5] `http://lists.grok.org.uk/pipermail/full-disclosure/2005-December/040442.html`
[6] For example: CVE-2007-1069, CVE-2007-1337, CVE-2007-1877 (each with a high CVSS rating of 7.8), and CVE-2007-1876 (with a high CVSS rating of 7.2)

6 Securing Virtualized IT Infrastructures

Virtualized IT infrastructures are not secure by default. We discuss basic aspects for providing security for virtualized IT infrastructures – by not claiming completeness of the subsequent listing. Our advices address general security measures. Thereby, we follow a local-to-global view onto the layers of a virtualized environment, starting with VMs, to virtual machine emulators, to network configurations to the overall IT infrastructure.

6.1 Providing Security for Virtual Machines

- ☞ Guest operating systems must be treated just as usual operating systems running on client or server computers. That is, they have to be hardened (e.g., disabling unneeded services), reliably configured and kept up-to-date.
- ☞ Emulated hardware and proprietary protocols that are not needed have to be disabled. For example, if possible, proprietary guest-to-host communication should be replaced by traditional network channels.
- ☞ Unpatched applications or operating systems within a VM are only allowed if there are valid reasons, the security implications have been examined and proper measures countering security risks are established (e.g., with respect to the network integration and the overall architecture).
- ☞ Make sure that rollbacks to earlier VM snapshots will not lead to vulnerable states, e.g., because of missing patches, changed configurations, or the reactivation of deactivated accounts.
- ☞ Guest operating systems need to be provisioned with self-contained security functionality – e.g., personal firewalls, antivirus software – rather than relying on the host only.

6.2 Providing Security for Virtual Machine Emulators

- ☞ In analogy to securing VMs, the host operating system has to be configured and maintained according to corporate information security guidelines.
- ☞ Virtual machine emulators have to be updated and patched to fix known flaws.
- ☞ To counteract VM-based rootkits [10,11,12], hardware-assisted virtualization (Intel VT-i, Intel VT-x, or AMD-V) should be
  - deactivated if virtualization is not used at all
  - consistently be used from the beginning on, otherwise.
- ☞ If provided, security features of the host operating system should be employed. For example, secure level features of BSD systems or the User Account Control (UAC) and Mandatory Integrity Control features of new Windows versions should be used. This helps to reduce the possibilities for attackers to elevate their privileges [13].

6.3 Providing Network Security

- ☞ If possible, shield VM's from network attacks by using host-only or NAT network configurations.
- ☞ If using bridged networking, make sure that all rules for network access that hold for normal hosts are followed for VMs, too.
- ☞ If network authentication and authorization is required for VMs, VMs need to be equipped with their own identifying *physical* network adapters so that network security as described by the IEEE 802.1x standard can be applied.

6.4 Providing Security for the Overall Computing Architecture

- ☞ VMs and the hosting physical machine must not belong to different security domains, except there are valid reasons backing this decision. That is, the security implications have been examined and proper additional security measures have been applied.
- ☞ Access to images of VMs on which sensitive data is stored or processed has to be protected. Possible threats are theft of image files or persistency of information that was not deleted due to snapshot functionality.
- ☞ A service hosting plan/contingency plan has to be defined in order to ensure availability.

6.5 General Security Management

The previous discussion showed that virtualization brings about several aspects which deviate from traditional IT processes. This is mainly due to the flexibility and ease of setting up and distributing images of complete systems. Thus, new processes are required to cope with such flexible infrastructures. Good practices should include

- clearly defined processes for the registration/provisioning/deregistration of VMs regulating responsibilities, (i.e., who creates/administers/uses VMs), license management, patch management, and storage of VMs, and
- guidelines for the integration of VMs into the network defining supported/prescribed network configurations (e.g., prerequisites for combining IEEE 802.1x and virtualization) and handling the identity of VMs (e.g., through registering dedicated MAC addresses for VMs).

7 Challenges and Advances

In Sections 4 and 5, security risks of virtualized infrastructures have been discussed. While Section 6 addresses rather conventional approaches for securing virtual machines, we highlight some interesting developments aiming at securing specific aspects of VMs, namely the safeguarding of VMs against data loss, the patching of guest systems and the inspection of intra-host communications.

7.1 Data Loss Prevention

In the previous sections we discussed several benefits of the isolation capability of virtual machines, like the possibility to execute untrusted code within a VM without the host being in danger of getting compromised. In contrast, a more or less reverse approach would be to safeguard VMs from getting compromised, e.g., by means of unallowed usage. For example consider a scenario where users are able to use a VM by executing applications provided by it or even to manipulate data stored on it. However, they should not be allowed to print information or to extract sensitive system states by writing data on network locations or on mobile storage devices like USB disks. That demands for VMs to be run in constrained environments which are controlled by a security policy. Such a configuration allows to improve control over sensitive data and applications, e.g., in cases when external employees require restricted access to corporate data.

Such requirements are, for example, addressed by VMware ACE [T10] which is a specialized version of VMware Workstation. VMware ACE allows you to distribute and install sensitive systems on machines which are not controlled by a security administrator. For that purpose, systems are packed in the form of VMware ACE images which can be run on dedicated host machines, whereby the distribution is controllable via digital rights management and their usage is configurable through security policies.

7.2 Inline Patching

One drawback of virtualized environments is that it is becoming easier to create new servers that don't adhere to corporate security policies. For example, guest operating systems might not be patched or even applications which are no longer supported by vendors are installed. Apart from such inappropriate usages of VMs, software which is maintained according to the company's security policies might still suffer from zero-day vulnerabilities, i.e., vulnerabilities for which patches do not yet exist.

Inline patching describes the process of fixing known vulnerabilities from the outside of a VM without actually patching the affected software itself. That is, patches are reverse engineered and the corresponding safeguarding functionality is provided through the virtualization layer. For example, if the vulnerability would be that the system crashes (e.g., due to buffer overflows) in case it receives malformed TCP/IP packages, the inline patch should detect and discard malicious messages before reaching the VM. This approach is more sensitive compared to the proceeding of a traditional IPS as it suffices to truncate malicious content instead of blocking the whole connection. The latter one would, in particular be detrimental for pooled connections like connections between databases and web applications [14,15]. This kind of security functionality is, for example, provided by Blue Lane's VirtualShield [T11] and Determina's LiveShield (now belonging to VMware). Both build upon reverse-engineered patches to protect vulnerable systems

7.3 Controlling Intra-host Communications

Traditional security appliances like external firewalls that filter traffic between physical network endpoints are unable to inspect intra-host communications, i.e., interactions between VMs where packets never leave the physical host. Security threats arising from uncontrolled guest-to-host or guest-to-guest communication include:

- Spread of malware and spyware over legitimate intra-host channels;
- Backdoors, i.e., unauthorized intra-host communications;
- Intra-host denial of service attacks;
- Intra-host spyware applications, e.g., interception of keyboard inputs, unencrypted IP communications, and file transfers.

Therefore, it would be beneficial to inspect intra-host communications, too. Novel fields of application are the integration of firewall, intrusion detection, or intrusion prevention functionality into virtualized environments. Thus, through packet inspection and content analysis threats and unwanted events can be identified.

Note that this usually would demand for a software solution, i.e., software which is installed on the physical host providing the virtualization infrastructure. This approach can provide cost and deployment advantages. However, disregarding usage scenarios where different networks/VMs which belong to different security levels reside on the same physical hosts (which is not recommended), it represents a configuration where security functionality like firewalls and IPS are combined on the same machines. However, hardly any alternative exists for virtualized environments. Thus, products like Reflex Virtual Security Appliance (VSA, [T12]) going into this direction actually "are focusing on virtualization as a solution to security problems, rather than just another attack vector" [16].

8 Conclusion

Virtualization is used for enabling resource partitioning, resource pooling, and for executing multiple operating systems or conflicting applications on one physical machine – concepts which are clearly justifying the use of virtualization as they provide benefits like reduction of total cost of ownership, flexible service allocation scenarios, and ease of administration.

Instead of focusing on operational benefits, this report discusses the security pros and cons of virtualization. It should have become clear that virtual machines are not primarily designed for security. That is, virtualization does not equal security! – although such arguments might turn up in advertising brochures. There are security risks, scenarios and vectors that are unique to virtualization software and architectures which must be considered very carefully. Therefore, it is high time to accompany the recent hype of bringing virtualization into data centers and to the desktop with the development of adequate security measures and processes. Announcements like the release of Microsoft's Hyper-V for Windows Server 2008 and VMware's VMsafe – an API enabling novel security approaches for virtual environments – are likely to give additional impetus in this direction.

List of Referenced Tools and Virtualization Products

[T01] QEMU
Project page: http://fabrice.bellard.free.fr/qemu

[T02] Microsoft Virtual PC for MAC
Product page:
http://www.microsoft.com/mac/products/virtualpc/virtualpc.aspx

[T03] Hercules
Project page: http://www.hercules-390.org

[T04] Bochs
Project page: http://bochs.sourceforge.net

[T05] Microsoft Virtual Server
Product page:
http://www.microsoft.com/germany/virtualserver/default.mspx

[T06] VMware virtualization products
Vendor page: http://www.vmware.com

[T07] Xen Source
Project page: http://www.xensource.com

[T08] VMDetect
Project page: http://www.codeproject.com/system/VmDetect.asp

[T09] Scoopy Doo
Project page:
http://www.trapkit.de/research/vmm/scoopydoo/index.html

[T10] VMware ACE
Product page: http://www.vmware.com/products/ace

[T11] Blue Lane VirtualShield
Product page: http://www.bluelane.com/products/virtualshield/

[T12] Reflex Virtual Security Appliance (VSA)
Product page:
http://www.reflexsecurity.com/products/reflexvsa.php

References

1. A. Perilli. *Step-by-step Virtualization: Using Virtualization to Improve Security*. January 2006. http://searchservervirtualization.techtarget.com/tip/1,289483,sid94_gci1188552,00.html.
2. T. Garfinkel and M. Rosenblum. *When Virtual is Harder than Real: Security Challenges in Virtual Machine Based Computing Environments*. In *Proceedings of the 10th conference on Hot Topics in Operating Systems (HOTOS'05)*, pages 20–20. USENIX Association, Santa Fe, NM, USA. 2005.
3. P. Ferrie. *Attacks on More Virtual Machine Emulators*. Technical report, Symantec Research. February 2007.
4. K. Lamb. *Virtualization and Security (IBM Internet Security Systems blog)*. September 2007. http://blogs.iss.net/archive/virtblog.html
5. T. Liston and E. Skoudis. *On the Cutting Edge: Thwarting Virtual Machine Detection*. In *SANSFIRE'06 Conference*. Washington, DC, USA. July 2006.
6. M. Carpenter, T. Liston and E. Skoudis. *Hiding Virtualization from Attackers and Malware*. IEEE Security and Privacy, volume 5(3):pages 62–65. 2007.
7. D. Quist and V. Smith. *Detecting the Presence of Virtual Machines Using the Local Data Table*. Technical report, Offensive Computing. March 2006.
8. T. Ormandy. *An Empirical Study into the Security Exposure to Hosts of Hostile Virtualized Environments*. Technical report, Google, Inc. February 2007.
9. D. Webber. *Can Virtualization be Trusted for Security?*. April 2006. http://advosys.ca/viewpoints/2006/04/virtualization-insecurity/
10. J. Rutkowska. *Subverting Vista Kernel For Fun And Profit*. In *Symposium on Security for Asia Network (SyScan '06)*. Singapore, China. July 2006.
11. S. T. King et al. *SubVirt: Implementing Malware with Virtual Machines*. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy (SP'06)*, pages 314–327. IEEE Computer Society, Washington, DC, USA. 2006.
12. D. A. D. Zovi. *Hardware Virtualization Rootkits*. In *Black Hat USA 2006*. Las Vegas, USA. August 2006.
13. Microsoft Corporation. *User Account Control Overview*. http://technet.microsoft.com/en-us/windowsvista/aa906021.aspx. October 2006.
14. A. Baumstein. *A Look at Blue Lane VirtualShield*. Technical report, University of Florida RealWorld Labs. May 2007.
15. L. Greenemeier. *Fighting Security Ghosts in the Virtual Machine*. March 2007. http://www.informationweek.com/story/showArticle.jhtml?articleID=198001244.
16. J. Hernick. *Virtualization Security Heats Up*. September 2007. http://informationweek.com/news/showArticle.jhtml?articleID=201803212