Virtualization and Security Aspects: An Overview

Rui Filipe Pereira *rui.pereira@protonmail.ch*
Lab UbiNET – Computer Science Security and Cybercrime
Polytechnic Institute Of Beja
Beja, Portugal

Rui Miguel Silva *ruisilva@acm.org*
Lab UbiNET – Computer Science Security and Cybercrime
Polytechnic Institute Of Beja
Beja, Portugal

João Pedro Orvalho *mail@joaoorvalho.com*
Lab UbiNET – Computer Science Security and Cybercrime
Polytechnic Institute Of Beja
Beja, Portugal

Abstract

Virtualization allows a single system to concurrently run multiple isolated virtual machines, operating systems (OSes) or multiple instances of a single OS. It helps organizations to improve operational efficiency, reduce costs, improve the use of hardware, and to allocate resources on-demand. Nevertheless, like most technologies, it has vulnerabilities and threats. Research about security issues related to virtualization has been conducted for several years. However, there are still open challenges related to security in virtualization. This paper looks into some of the differences, issues, challenges, and risks caused by virtualization and aims to classify the various virtualization approaches, along with their goals, advantages and drawbacks from a security perspective. Such classification is expected to help in the identification of virtualization technologies that might be applied in a virtualized infrastructure.

This work is intended to be an introduction to the security considerations, concerns, and implications arising from use of virtualized systems.

Keywords: Virtualization, Cybersecurity, Hypervisor, Virtual Machine, Virtual Machine Monitor.

1. INTRODUCTION

Virtualization is a technology that uses a logical environment to overcome the physical limitations of hardware. Due to its characteristics of encapsulation and isolation, virtualization is the basis for cloud computing paradigm. It can be characterized as a complex technology with many facets and numerous types of controls that can be implemented to protect virtual assets and their host's machines. It is an opportunity, but also a threat [1].

Some studies have shown that users who are planning to migrate to cloud computing are considering security as one of the most important factors [2]. Past events have shown that software vulnerabilities are unlikely to stop, and virtualization is also prone to these traditional software vulnerabilities (e.g. buffer overflow vulnerability). Furthermore, the design, implementation and deployment of virtualization technology has revealed new threats and security issues that, although not specific to virtualization, take new forms in relation to it [3].

In [4], was showed how insecure virtual hosts may be, due to poor virtualization software designs and code. Several efforts have been made to address virtualization software security issues

through the use of sandboxes [5], or by measuring the runtime integrity of virtualization software components [6].

Current OSes provide an abstraction of processes to achieve resource sharing and isolation. However, from a security perspective, an attacker who compromises a process may gain full control over the system [7]. This makes the security systems that are running on the same system, such as anti-virus programs and intrusion detection systems, also vulnerable to attacks.

In response to the imperfect isolation among processes, virtualization can be used to ensure the confidentiality and integrity of information. Secure isolation is one of the most important concepts of virtualization [4]. Managing the security of a physical machine can usually be seen as familiar, using well-known procedures as this has been done for many years. However, with virtualization on a single physical machine can be multiple OSes, multiple network interfaces and hundreds of applications or services, making the technological infrastructure increasingly complex and heterogeneous.

Virtualization does not mean security or replacement of security. In fact, virtualization brings a more complex and risky security environment to manage [8].

This is an evolving technology that adds more complexity to the already difficult path to compliance due to the strict requirements of standards and regulations [9]. With virtualization, virtual servers commonly have multiple functions, such as databases and web servers, running on a single physical server. However, according to the section 2.2.1 of Payment Card Industry Security Standards Council (PCI DSS), a server should perform only one primary function per server, which goes against the goal of virtualization technology (to promote the consolidation of multiple services on a single server) [10].

Virtualization presents a challenging topic that combines different software and/or hardware-based technologies to create an abstraction layer.

The distinct types of virtualization technologies, and security implications on virtualized infrastructures will be presented throughout this paper as well as a detailed analysis of attack strategies that can be used against virtualization infrastructures. Plus, a set of general recommendations is provided to achieve safer virtualized implementations.

This study aims to classify various virtualization approaches, goals, advantages and drawbacks from a security perspective. The work also present attack vectors, security risks in virtualized infrastructures, and various possible threats associated with virtualized environments. It is also proposed a set of general recommendations to achieve safer virtualized implementations.

In this paper, an extensive literature search has been conducted. The literature is retrieved from well-known sources such as leading journals and additional literature is found by tracing back the cited papers and forward towards conferences papers. Literature from other sources was also included, such as datasheets from virtualization product vendors such as VMware. Major publications from the literature have been grouped and studied, as it allows the analysis and discussion of multiple aspects of virtualization and security concerns.

The remainder of this paper is organized as follows. The next section discusses virtualization components, Section 3 addresses the classification of virtualization technologies, Section 4 clarifies security terms and definitions, Section 5 mentions attack vectors and security threats in virtualized infrastructures, and in the Section 6 several attacks and security risks in virtualization infrastructures are mentioned. Finally, Section 7 mentions security considerations and Section 8 presents the conclusion.

2. VIRTUALIZATION COMPONENTS

In this section are described components related to virtualization.

2.1 Hypervisor

Also known as Virtual Machine Monitor (VMM) [11]–[16], it is the main component of a virtualization system and keeps track of activities carried out by virtual machines (i.e., it manages VM applications), forwards hardware requests to physical resources, provides replicated platforms, and supports resource sharing among different virtual machines.

It provides an abstraction layer to virtualized systems, thus emulating hardware devices for each virtual machine and making virtual communications available between VMs and physical resources, acting as a mediator between virtual machines and the underlying physical devices.

There are two types of hypervisors, namely: bare-metal and hosted [17], [18]:

- **Type 1 (bare-metal):** also known as native. Hypervisor runs directly on the system hardware (e.g. Vmware ESXi, Xen);
- **Type 2 (hosted):** hypervisor runs, as an application, on a host OS that provides virtualization services (e.g. Oracle VM VirtualBox, VMware Workstation).

Hypervisors of type 1 are mainly OS that boot with the system and are used as virtualization servers. The security of a virtualization system is based on the security capabilities of the VMM.

Based on the architecture, the type 1 can be classified in two models, microkernel and monolithic:

- **Monolithic:** device drivers are included in the hypervisor core, providing a better performance as communications between hardware and software dot not require any intermediate. However, as a consequence, the hypervisor requires more lines of code, increasing the attack surface;
- **Microkernel:** device drivers are installed on the OS of the guest machine, reducing the footprint of VMM. Moreover, communication between software and hardware is mediated by the VMM, leading to a better security but a worse performance.

In type 2, the host's OS is responsible for managing and providing the I/O of the virtual machines, adding another layer of abstraction [17], [18].

From a security perspective, there is a strong divergence between type 1 and type 2 hypervisors, because the attack surface is considerably larger in the type 2 hypervisors since the OS where the hypervisor is installed is a whole surface that can be attacked.

A VMM is responsible for performing two essential tasks: enforcing isolation between VMs and managing the underlying hardware resources.

All interactions between VMs and hardware must go through the VMM. Any hosted VM must be prevented from accessing parts of the memory that belong to another VM, similarly, that a potential failure in a VM should not interfere with the normal behavior of other VMs. To provide isolation and minimize the consequences of errors in the software, VMM uses the Memory Management Unit (MMU) as well as other hardware units.

The hypervisor should manage CPU load balancing, map physical addresses to logical memory addresses, migrate VMs between physical systems and so forth, while protecting the integrity of each VM and protecting the stability of the entire system [19].

Hypervisors should be as minimal and light as possible to achieve efficiency and "optimal" security. Hypervisors are considered more secure than OSes in general [20].

2.2 Guest Machine

Also known as virtual machine, it instantiates the virtualized (encapsulated) system composed by the OS and applications, using the hardware abstraction provided by the VMM. Guest machines are isolated by the hypervisor, which controls their activities, and behave as if they were in a single execution environment with their own dedicated resources. Each guest machine can install a different OS to support virtualization heterogeneity [17].

2.3 Host Machine

Host machine is the real physical machine and its OS (host OS) that hosts the virtualized environment. The host OS directly manages the underlying physical hardware, the virtualized environment, and is where the hypervisor is operated. Sometimes the term "host OS" also refers to the privileged VMs, which, in specific virtualization approaches, support the operation of the virtual machines (e.g., providing a set of drivers to facilitate access to the underlying physical hardware) [17].

2.4 Management Server

It is the virtualization platform composed of a set of components for directly managing the virtual machines, consolidating services, allocating resources, migrating virtual machines, assuring high availability, among others [17].

2.5 Management Console

It is the component that provides access to a management interface to the virtualization system for configuring and managing virtual machines. Virtual machines can thus be added, modified, deleted or configured. It can be provided as a standalone client or via a web interface to visually handle management server functionalities.

Examples of management consoles include VMware vSphere client console and the VMware vSphere web client [17].

2.6 Network Components

They facilitate the development of virtual networks, where virtual network devices (e.g., switches, routers) are completely controlled through software and the network protocols and hardware are simulated. Virtual machines can be connected in the same way as physical machines and built on host-machine physical network infrastructure to connect to the public network [17].

2.7 Virtualized Storage

It provides all the components for abstracting physical storage in a single storage device which can be accessed either over the network or through a direct connection. Storage virtualization introduces additional management overhead, since stored data can be only logically partitioned in different storage locations while belonging to the same shared storage.

Storage virtualization can address many types of physical storage technologies, including direct attached storage (DAS), storage area network (SAN), and network attached storage (NAS).

Examples of these devices include RAID arrays hosted inside a server computer (DAS), storage devices collecting all datacenter data such as EMC VNX7500 (SAN), and a simple storage component that offers network file-level access through a wide variety of application protocols such as CIFS or NFS [17].

3. CLASSIFICATION OF VIRTUALIZATION TECHNOLOGIES

Virtualization technologies are classified according to their degree of hardware emulation and virtualization level. It is possible to distinguish between approaches that provide full hardware emulation and approaches that provide hardware virtualization (or OS virtualization or partial hardware emulation).

Besides system-level virtualization (hardware emulation, full virtualization, paravirtualization, hardware-assisted virtualization) where virtualization is at the granularity of a virtual machine executing a complete system, it is also described in this section two further classes, namely OS-level virtualization and application-level virtualization.

Table 1 summarizes the comparison of virtualization types in Hypervisors.

	Full Virtualization with Binary Translation	**Hardware Assisted Virtualization**	**OS Assisted Virtualization/ Paravirtualization**
Technique	Direct Translation and Execution	Root mode in privileged instructions	Hypercalls
Modification of the Guest OS and Compatibility	It is not necessary to modify the Guest OS; Excellent Compatibility	It is not necessary to modify the Guest OS; Excellent Compatibility	Guest OS is modified for Hypercalls, so it does not run on Native Hardware or another Hypervisor; Poor Compatibility; Not available on Windows systems
Performance	Good	Fair	Better in some conditions
Used by	VMware, Microsoft, Parallels	VMware, Microsoft, Parallels, Xen	VMware, Xen

TABLE 1: Comparison of virtualization types in Hypervisors.

3.1 Full Virtualization with Binary Translation

Full virtualization supports virtualization of x86 systems by simulating the underlying hardware. The hardware is simulated in software by each virtual machine.

Full virtualization can be based on a mix of binary translation of kernel code and direct execution of user-level code.

The guest OS runs unmodified with no need for hardware or OS support. With binary translation the OS does not need to be aware that virtualization software runs on the system. The underlying hardware and the guest OS are fully abstracted and separated, and within them lies the virtualization layer provided by the VMM.

Binary translation transforms and caches the kernel code that needs to be executed by the guest OS. Full virtualization provides a solution with highest isolation and security, while it decreases performance and adds more overhead as instructions are translated in real time [17], [19], [21].

Figure 1 illustrates this approach.

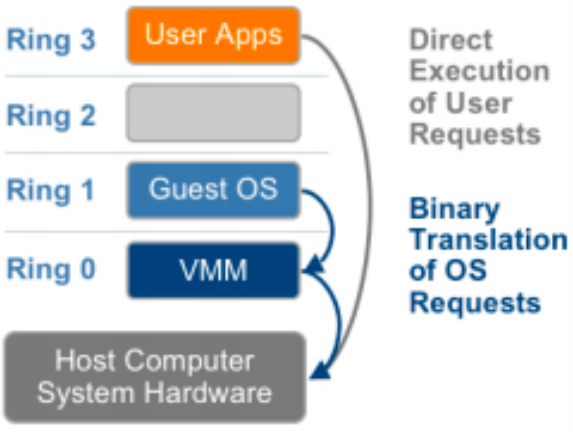

FIGURE 1: Binary translation approach to x86 virtualization [22].

The hypervisor, in ring 0, is responsible for capturing privileged instructions that could not be virtualized, translating them and replacing them with new instructions that have an effect on virtual hardware.

The fact that the OS does not need to know that it is being executed by the complete virtualization technique with binary translation is an advantage as several kernels of OSes cannot be modified.

3.2 OS Assisted Virtualization or Paravirtualization

It is a technique that modifies the kernel of the hosted OS, not requiring binary translation.

VMM has built-in software that presents an appropriate interface for hosted virtual systems, such as drivers to interact directly with the hardware.

It provides a lightweight virtualization technique where the hypervisor exposes hypercalls that can be directly called by a modified guest OS to simulate privileged instructions that are difficult to virtualize. The hypercalls implement a virtualized version of system calls and invoke the hypervisor's services. They can be called by a modified guest OS through known APIs.

Paravirtualization provides better performance and lower overhead than full virtualization as it does not require emulation of system resources. The performance of paravirtualization over full virtualization with binary translation is significantly better in several configurations, and for some workloads it is close to native [17], [19], [23]–[26]. However, due to the prices associated with modifying proprietary software, it is normal to see only modified open source guest OSes.

3.3 Hardware-assisted Virtualization

Although paravirtualization increases the performance, it cannot be as great as native virtualization, since it involves the mediation of the driver interface to allow interaction between virtual machines and hardware.

Hardware vendors are rapidly embracing virtualization and developing new features to simplify virtualization techniques. First generation enhancements include Intel Virtualization Technology (VT-x) and AMD-V, and both target privileged instructions with a new CPU execution mode feature that allows the VMM to run in a new root mode below ring 0.

As shown in Figure 2, privileged and sensitive calls directly trap the hypervisor, requiring neither binary translation nor paravirtualization.

The guest state is stored in Virtual Machine Control Structures (VT-x) or Virtual Machine Control Blocks (AMD-V). Processors with Intel VT and AMD-V became available in 2006, so only some systems contain these hardware assist features [27].

A new CPU state was introduced, orthogonal to privilege rings 0-3, called root mode on Intel chips and guest mode on AMD chips. This state is accessed whenever the hypervisor needs to take the control over a virtual machine. The guest OS can run in ring 0 but not in root mode.

Furthermore, the handling of I/O memory virtualization allows to prevent Direct Memory Access (DMA) requests issued from a virtual machine to tamper with unauthorized zones of the host memory.

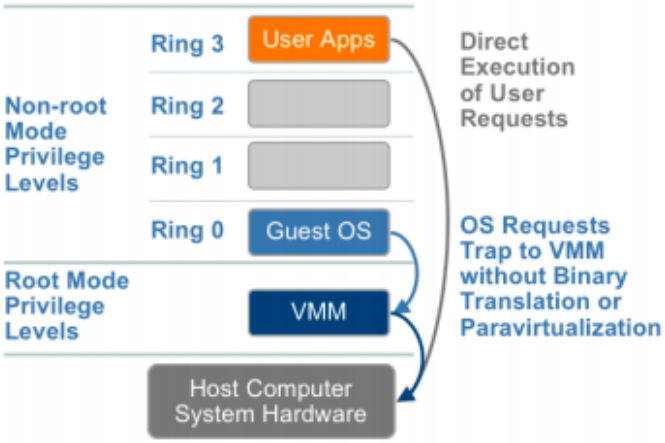

FIGURE 2: Hardware-assisted approach to x86 virtualization [22].

3.4 Full Hardware Emulation

It allows executing an unmodified system (guest OS) in a different host architecture.

Emulation, such as full virtualization, is compatible with unmodified OSes. However, in that case, the resources seen by the guest OS are completely simulated by software. This allows to execute an OS compiled on a different architecture from the architecture of the host.

Examples of solutions that support full hardware emulation include Bochs, QEMU and VirtualPC [5].

3.5 Operating-system-level Virtualization

It is based on an OS that supports multiple instances of isolated user-space, called containers. Each container can target a single application and install only the required software and libraries to run an application.

The host machine's hardware resources are partitioned among different guest machines. The host OS deploys many instances of guest OSes, with a lightweight execution of the OS or application.

Resources are assigned to containers that represent a set of processes, files, and partitions. This approach provides high performance, low overhead, and allows the execution of the same OS as the host machine. Examples of solutions supporting OS-level virtualization include Docker, Virtuozzo, OpenVZ, and Solaris Containers.

3.6 Application-level Virtualization

This approach increases programs' portability among different software-hardware architectures. It is based on various components, including: a portable language, a compiler between source code and an architecture-independent representation (bytecode), a bytecode interpreter, and an execution environment that translates bytecode into low-level operations on the host machine. Examples of solutions supporting application-level virtualization include Java VM, Microsoft .NET, Perl, Python, and Ruby [5], [17].

4. CLARIFICATION OF SECURITY TERMS AND DEFINITIONS

In this section it is provided an overview of several security terms and definitions as it is necessary to clarify the terminology adopted, namely:

- **Threat:** any circumstance or event with the potential to adversely impact an asset through unauthorized access, destruction, disclosure, modification of data or denial of service (DOS);
- **Threat agent:** someone or something with some capacity, a clear intention to manifest a threat, and a record of past activities in this regard;
- **Weakness:** a type of mistake in software, in operations and in the infrastructure, that, in the right conditions, could contribute to introducing vulnerabilities. This term applies to mistakes in software, regardless of whether they occur in implementation, design or other phases of the software-development life cycle;
- **Vulnerability:** an occurrence of a weakness (or multiple weaknesses) in software, operations or infrastructure, in which the weakness can be used to perform actions that were not specifically granted to who takes advantage of the weakness. So, vulnerability must always be described in terms of resistance to a certain type of attack [28];
- **Impact:** the effect of an event, incident or occurrence. In cybersecurity, this means the effect of a loss of confidentiality, integrity or availability of information on an organization's operations, an organization's assets, individuals, other organizations or national interests. The potential impact (severity impact) of weaknesses and vulnerabilities on organizations can be measured in qualitative terms as low, moderate, and high;
- **Risk:** a function of the likelihood of a given threat source exercising a potential vulnerability, as well as the resulting impact of that adverse event on an organization. Risk can be given by: Risk = Probability · Impact.

5. ATTACK VECTORS AND SECURITY ISSUES IN VIRTUALIZED INFRASTRUCTURES

In this section is mentioned common weakness in virtualized environments. It is presented the weaknesses in general and we will detail them in the virtualized infrastructure based on the Common Weakness Enumeration (CWE) maintained by MITRE.

5.1 Injection

This weakness is based on the lack of verification of user-controlled input, or improper accessing data without proper authorization.

In virtualized environments, injection issues still exist on interaction-specific virtualization. They are often not well-tackled since the users involved frequently have administration-level permissions. A specific type of injection is VM image/VM template injection.

Among the related CWE are: Injection (CWE-74), Code injection (CWE-94), OS command injection (CWE-78), SQL command injection (CWE-89), CRLF injection (CWE-93), and Externally-controlled format string (CWE-134) [9], [17].

5.2 Improper Authentication

Authentication is a part of the security model Authentication, Authorization and Accounting (AAA).

It is a process by which the system or application validates supplied credentials and assigns appropriate privileges. When an actor claims to have a given identity, and the software does not prove or insufficiently proves that the claim is correct this can lead to the exposure of resources or functionalities to unintended actors. This is caused by incorrectly designed or implemented authentication mechanisms.

In virtualized environments, authentication applies both to end users and to system components. Examples of these weaknesses include the use of inappropriate credential types or verification mechanisms, such as using password-based authentication instead of certificates in highly volatile and dynamic environments or using weak registration mechanisms or bugs in the authentication processes.

Among the related CWE are: Improper Access Control (CWE-284), Improper authentication (CWE-287) Improper authorization (CWE-285), Incorrect user management (CWE-286), Placement of user into incorrect group (CWE-842), and Improper restriction of the communication channel between the endpoints (CWE-923) [10], [17], [29].

5.3 Management of Credentials

One authentication mechanism is only as strong as how its credentials are managed. Due to this reason, it is important to require users to have strong passwords, and to enforce password strength.

The lack of password complexity significantly reduces the search space when trying to guess user's passwords, making brute-force attacks easier.

This weakness also refers to insufficiently protected credentials, both at storage and in transit (i.e., plaintext storage or unprotected transport).

Virtualized environments exacerbate this weakness group because they share unprotected transportation channels, incrementing the number of actors that may be able to sniff credentials. In virtualized environments, this may affect multiple levels of the virtualization stack.

Among the related CWE are: Credentials Management (CWE-255), Weak cryptography (CWE-261), Weak password recovery mechanism for forgotten password (CWE-640), Insufficiently protected credentials (CWE-522), and Hard-coded credentials (CWE-798) [17], [30].

5.4 Permissions and Privileges Management

Weaknesses in this category are related to the management of permissions, privileges, and other security features that are used to perform access control. Specifically, it includes issues caused by the execution without the required or incorrect privilege assignment, errors in increasing or diminishing privileges, and insecure or preserved inherited permissions.

In virtualized environments, this weakness is emphasized by the complexity of the privileges and multiplicity of administrative layers needed for a virtualized environment, especially considering its dynamics, and scenarios where migrations and federations are in place.

Among the related CWE are: Permissions, Privileges and Access Control (CWE-264), Privilege and sandbox Issues (CWE-265), and Permission issues (CWE-275) [17], [31].

5.5 Cryptographic Issues

This kind of weakness is related in particularly to cryptographic errors caused by poor design or poor implementation of the cryptographic solution, or weaknesses in cryptographic protocols by missing or weak encryption of sensitive data during storage or transmission, allowing man-in-the-middle (MITM) attacks.

Other example can be plaintext storage or transmission of sensitive information, key-management errors such as key exchange without entity authentication and lacking or weak verification of expired keys.

Virtualized environments exacerbate cryptographic issues by sharing of channels or resources. MITM attacks become highly critical in virtualized environments, where messages from different users may share the same channel or infrastructure facilities.

Among the related CWE are: Cryptographic Issues (CWE-310), Key management errors (CWE-320), Missing encryption of sensitive data (CWE-311), and Missing required cryptographic step (CWE-325) [17], [32]–[34].

5.6 Data Handling

Weaknesses in this category are typically found in the functionality that processes data. It is a broad category as it includes string and type errors, generic representation errors such as improper handling of syntactically invalid structure, and numeric errors (e.g., wrap-around error or incorrect conversion between numeric types).

In virtualized environments, this also involves data-remanence issues, which are typical of virtualization and exacerbated by shared storage or memory resources.

Among the related CWE are: Data handling (CWE-19), Representation errors (CWE-137), and Numeric errors (CWE-189) [17], [35], [36].

5.7 Information Management Errors

This refers to weaknesses that involve improper handling of sensitive information. It specifically includes information exposure or information leak in intentional or unintentional way to an unauthorized actor.

In virtualized environments, attacks that exploit this weakness are more critical than in physical environments. In addition, the distribution and replication mechanisms that belong to such environments facilitate data-mining attacks. Finally, covert channels that exploit physical CPU architecture become more critical due to CPU and memory sharing, which permits extraction of information about processes or networking traffic that belong to other users.

Among the related CWE are: Information management errors (CWE-199) and Information Exposure (CWE-200) [17], [18], [37].

5.8 Improper Input Validation

When software does not validate input properly, an attacker is able to craft the input in a form that is not expected by the rest of the application. This will lead to parts of the system receiving unintended input, which may result in altered control flow, arbitrary control of a resource, or arbitrary code execution. This refers to pathname traversal and similar issues, including improper link resolution before file access (link following). It also includes memory-buffer weakness such as classic buffer overflow and out-of-bound read or write issues.

In virtualized environments, the stratification of interacting software components increases the impact of this weakness, and verification becomes difficult due to the complexity of the interactions at component levels. In addition, referring strictly to user interaction, this weakness shares the same issues as the injection-weakness group.

Among the related CWE are: Improper Input Validation (CWE-20), Path traversal (CWE-22), Link Following (CWE-59), Memory buffer (CWE-119) [17], [38], [39].

5.9 Insufficient Verification of Data Authenticity

This class of weaknesses is a result of trust issues between data exchange parties.

If an application fails to verify data origin or its authenticity, an attacker might be able to perform spoofing attacks against a vulnerable application or its clients.

Lack of data authenticity verification may arise in a variety of situations and most likely to be introduced at design and implementation stages of application development process.

It can include improper verification of cryptographic signature, missing or improper validation of integrity check and Cross-Site Request Forgery (CSRF).

Virtualization supports technology such as Intel-VT, secure crypto-processors, and Trust Computing (TC/TPM), which provide fundamental virtualization features but also open a set of virtualization-technology-specific weaknesses (e.g., hypervisor blue-pilling rootkit in nested virtualization or misbehavior in authenticity-verification during boot). Weaknesses in the cryptographic-issues group may also underlie insufficient verification of data origin and authenticity.

Among the related CWE are: Insufficient Verification of Data Authenticity (CWE-345), Cross-Site Request Forgery (CWE-352), and Improper verification of cryptographic signature (CWE-347) [40], [41].

5.10 Improper Certificate Validation

When a certificate is invalid or malicious, it might allow an attacker to spoof a trusted entity by using a MITM attack.

A software might connect to a malicious host while believing it is a trusted host, or a software might be deceived into accepting spoofed data that appears to originate from a trusted host [42]. Therefore, this weakness is related to improper validation with host mismatch, certificate expiration, revocation or missing validation. It also includes weaknesses related to improper following of certificate's chain of trust.

In virtualized environments, this weakness is exacerbated by the fact that the confidentiality and integrity of (both internal and external) communication between virtualization components when is based on certificates, while certificate protection is at stake due to sharing and the multitenant nature of the virtualization infrastructure. Improper certificate validation can then result in unprecedented consequences and impacts.

Among the related CWE are: Improper Certificate Validation (CWE-295), Certificate expiration (CWE-298), Check on revocation (CWE-299), and Missing validation (CWE-599) [17].

5.11 Use of Insufficiently Random Values

This type of weaknesses involves generating predictable values in a context that requires unpredictability.

It is related to insufficient entropy in pseudo-random number generators (PRNGs), predictability problems, and the use of cryptographically weak PRNGs.

In virtualized environments, this weakness is exacerbated by the virtualization of hardware devices. For instance, achieving sufficient entropy is even more difficult since the virtualized environment reduces the quality of the source of entropy commonly adopted by PRNG algorithms.

A related CWE is Insufficiently Random Values (CWE-330).

5.12 Resource Management Errors

This type of weaknesses involves improperly managing system resources, possibly leading to resource exhaustion.

It also refers to weaknesses stemming from improper resource shutdown or release, double free call that leads to modifying unexpected memory locations, and many other memory-management weaknesses, such as the improper release of memory before removing the last reference.

In virtualized environments, this is crucial because several attacks are based on exhausting system resources to achieve DOS or to force the system into a state that facilitates other attacks.

Resource-consumption issues show a transversal impact on many components, from hypervisors, which may not be not able to offer balanced computing power, to virtualized networks, which may represent a serious bottleneck due to resource exhaustion.

A related CWE is Resource Management Errors (CWE-399).

5.13 Cross-site Scripting

This group refers to user-controllable input that is not neutralized or is incorrectly neutralized before it is placed in an output that is used and served to other users.

It is mainly for web pages. As a result, an attacker can inject and execute arbitrary HTML and script code in user's browser in context of a vulnerable website.

Based on weakness conditions it is common to divide XSS errors into 3 main types: reflected XSS, stored XSS and DOM-based XSS. After successful attack a malicious user can perform a variety of actions: steal user's cookies, modify webpage contents, perform operations with the site within user's session (XSS proxy).

In virtualized environments, there are dashboards to evaluate virtualization features or to inspect resources. These web-based dashboards allow interaction and thus must be protected against cross-site scripting [17]. A related CWE is Cross-site Scripting (CWE-79).

5.14 Race Conditions

This group refers to sequences that can run concurrently with other code, and the code sequence requires temporary, exclusive access to a shared resource, but there are time windows in which the shared resources may be modified by code sequences that operate concurrently.

This can have security implications when the expected synchronization is in security-critical code, such as recording whether a user is authenticated or modifying important state information that should not be influenced by an outsider.

A race condition occurs in concurrent environments and is effectively a property of a code sequence. Depending on the context, a code sequence may be in the form of a function call, a small number of instructions, a series of program invocations, etc.

It violates exclusivity (the code sequence is given exclusive access to the shared resource) and atomicity (the code sequence is behaviorally atomic) properties, which are closely related.

A race condition exists when an "interfering code sequence" can still access the shared resource, violating exclusivity.

In virtualized environments, the existence of numerous independently managed, asynchronous components mandates carefully designing and implementing mechanisms to manage such situations.

A related CWE is CWE Race Condition (CWE-362) [17].

5.15 Environment

This group refers to weaknesses introduced during unexpected environmental conditions. It refers mainly to technology-specific issues and interaction error occurred when two entities work correctly when running independently, but they interact in unexpected ways when they are running together.

In virtualized environments, several software components interact to bring virtualization facilities to the end users. This ecosystem is made up of software from different vendors that use different technologies, developed and maintained according to different methodologies. This emphasizes issues related to the coexistence and cooperation of software components in virtualization systems, as well as leading to the weakness group "configuration" [17].

Among the related CWE are: Environment (CWE-2) and Interaction Error (CWE-435).

5.16 Configuration

This group refers to weaknesses typically introduced during the configuration of the software components. Virtualized systems are often based on several interoperating software components that need to be dynamically configured to support virtualization in many application scenarios.

Weaknesses at the configuration level grow in importance when virtualization behavior is affected by dependencies among different components. In addition, all these components are based on complex configurations, which, due to the interactive nature of the components, may evolve during the virtualized-environment lifecycle. This makes weaknesses in the configuration group even more significant in virtualized environments than in traditional systems, because in virtualization the logical layer is more complex.

A related CWE is Configuration (CWE-16).

6. ATTACKS AND THREATS IN VIRTUALIZATION INFRASTRUCTURES

In this section, several attacks and security threats related to virtualization infrastructures are mentioned.

6.1 Lack of Security Controls

Segregation of systems, i.e. different systems for different purposes (e.g., production systems and development systems), is common. Due to their nature, systems for development may have fewer security controls in place, this may eventually provide an easier way for a possible intrusion.

Since VMs are not physical machines, they are all stored as a collection of files whether on the local hard drive or on another type of support (e.g., NAS, SAN).

If an attacker gets physical access to the hypervisor or the storage devices it may misuse an entire OS or download the virtual image to your system.

6.2 Malware

Virtualization is a powerful tool for deploying a virtual environment for malware analysis. However, there are techniques for detecting the presence of virtualization software [43]–[45].

If a system is detected as being virtual, malware that is aware that it is present in a virtual environment may change its behavior accordingly and intend to directly attack the VM and its components or attack the virtualization layer itself (VMM).

The most common way for a malware to infect a system is by exploiting vulnerabilities that are usually found in software.

Research has been conducted to find and protect the means by which malware detects a virtual environment (e.g. VMM issues, registry entries, OS peculiarities, or CPU indicators) [46].

Malware attacks can create possible situations for increased workload of the infected machine.

Examples of malware for virtual environments are SubVirt and Blue Pill.

6.3 Reversion of VM

Several encryption solutions are based on using the system configuration to generate a seed and create hashes. A seed can be obtained from system clock, hard disk rotation, memory contents and many other system elements. Besides that, seeds can be used to create timestamps or nonces. The rollback of a VM can lead to some seeds being used again, in the same way that they were used for previous communications to create timestamps and nonces.

A snapshot allows to create a full image of a client machine at a certain point in time. Although this feature is very useful, it can also bring security problems, namely:

- Insert into the network a machine that does not have the latest updates;
- Re-activate accounts that have been decommissioned;
- Use of old security policies.

6.4 VM Sprawl

VM Sprawl describes the situation where the number of VMs on a network goes beyond the point where they can be managed effectively.

It occurs when there is an uncontrolled implementation of virtual machines in productive environments, without managing changes in virtual machines, without a formal review process for the security of virtual machines before they are implemented, and without a restricted set of licensed VM models.

Without an effective control process in place, VMs and other virtual systems with unknown configurations can quickly proliferate, consuming resources, degrading overall system performance, and increasing liability and risk of exposure.

6.5 Memory Congestion

It is concerned with the allocation of resources, which include link bandwidth, memory size, and processing capacity at all intermediary nodes, among all the connected nodes in a network. The connection will be as such that the nodes can operate the transaction at an acceptable performance level.

However, the resource allocation is necessary, even for a low load, but the problem becomes challenging when the load increases. Due to a massive load on a single node, the fairness issues will occur, and low overhead will increase, which reduces the performance [47].

6.6 Hyperjacking

Security measures, such as firewalls, IDS/IPS, and antivirus are ineffective against hyperjacking because neither the VM nor the server is aware that the hypervisor is compromised. Two examples of Hyperjacking are Virtual-Machine-Based Rootkit bluepill and SubVirt.

6.7 VM Escape

The VMs are encapsulated, in isolated environments, and the OSes running inside the virtual machines should not know that they are running in a virtualized environment, nor should there be a possibility to leave the VM and change the hypervisor.

It is called VM escape when this isolation is broken, and the VM hosted, interacts with the hypervisor. In VM escape, a program that runs on a VM is able to bypass the virtual layer (provided by the hypervisor), and gain access to the host machine.

Several organizations compromise isolation by configuring it in a flexible way in order to meet the needs of the organization, leading to security issues. The solution of this vulnerability involves correctly configuring the host machine and VM client interaction.

6.8 Denial of Service

It can be described as an attack that happens when a hosted machine uses almost exclusively all available resources. Therefore, it is important to prevent a hosted machine from consuming all resources by limiting resource allocation to each VM.

6.9 VM Poaching/Resource Hogging

It is similar to DoS attack and happens when one of the virtual guest OSes takes up more allocated resources against another guest OS on the same virtualized environment.

This issue can make the virtual machine crash. VM poaching can occur in any resources of a hypervisor such as disks, memory, CPU, network, and disks.

6.10 Incorrect Isolation between VMs/VMs and Hosts

As mentioned, one of the key issues in virtualization is isolation. Isolation ensures that an application that is running on a VM cannot see applications running on another different VM, or that a process running on a VM cannot affect other VMs running on the same physical machine.

If this insulation is not working properly, then an attacker can access other virtual machines on the same machine or even the hypervisor.

In such a heterogeneous environment it is difficult to guarantee the operational integrity of each VM. The same is applied when there are hardware faults on the host machine that can affect the several hosted systems.

6.11 Intercommunication Among Virtual Machines

It is provided by virtual switches embedded in the VMM. These switches allow communication between VMs hosted on the same machine, using the same protocols that physical systems use, not requiring to install additional network interfaces.

The visibility of VMs' intercommunication is limited and monitoring connections or performing network diagnoses can be considered as a difficult task to accomplish. The main reason is that to monitor virtual switches, is required a robust and reliable subsystem in the hypervisor to provide statistics, flow analysis and problem-solving capability. Hypervisors usually lack extended features such as those to avoid heavy and complex implementations and to minimize security issues.

Unless the monitoring tools are in each VM, the lack of visibility poses a great danger to the environment itself.

6.12 OS Vulnerabilities

It is the OS that controls the way the computer runs each software. Therefore, a vulnerability in the OS can lead to serious security risks (e.g. attacker takes control of Administrator account).

6.13 External Hypervisor Modification

Unexpected hypervisor behavior can break the system’s security model.

There are several solutions to this problem, for instance to use technologies such as Secure hypervisor approach to Trusted Virtualized Systems (SHype) to ensure the security of the hypervisor layer [7].

Another solution is to protect the hypervisor from unauthorized modifications or to allow hosted VMs to validate the hypervisor [20].

6.14 External Modification of a VM

The best approach to solve this problem is to assign a digital signature to the VM and validate that same signature before the implementation.

6.15 Monitoring of a VM from Another VM

If correctly implemented, memory protections should not allow a VM from viewing the memory used by the other VM. However, if the virtualization platform uses a virtual hub/switch to connect all hosted VMs with the host, VMs may be able to capture packages (e.g. with ARP poisoning attack).

Network traffic authentication is considered a possible solution, and it is also be possible to limit the MAC Ethernet address that can be used in a VM virtual network interface. Nevertheless, MAC Address can be spoofed.

6.16 Monitoring of VMs from The Hypervisor

A major concern regarding the administration of a virtual infrastructure is the way in which several workloads hosted in a single physical host are managed.

In general, all network traffic to and from VMs goes through the hypervisor. This allows allows hypervisor to monitor all network traffic for all VMs.

If a hypervisor is compromised, then the security of VMs may also be compromised.

6.17 Attack Guest-to-Guest

Is assumed that the attacker has already gained access to a hosted VM.

These attacks are usually performed indirectly e.g. an unauthorized user escapes from a hosted environment and then compromise the other hosted VMs through privileged access to the hypervisor.

7. SECURITY CONSIDERATIONS

Traditional information security risks are inherited by virtualization technology and are added to the new ways and methods of executing and manipulating the security of a virtualized system.

Most of the information security standards mention the use of robust monitoring solutions with the ability to keep track of all changes that occur in a system or any other incident that may be useful for possible investigations.

Technological advancements have allowed the development of virtual machine introspection techniques. They replaced traditional methods of monitoring protection, which were inadequate in today's demanding and critical virtual environments.

The great complexity and extensive functionality of today’s systems highlight the tendency to become vulnerable to design errors or programming errors.

The larger the surface of an OS, the more likely it is to contain bugs or design errors. Therefore, one of the essential characteristics of hypervisors is that they must be as minimal and light as possible in order to achieve levels of efficiency and security very close to ideal.

8. CONCLUSION

Virtualization, as a technology, was able to ensure efficiency in infrastructure, as well as create the consolidation of a large number of services in a small number of is physical machines.

Hypervisors are having an increasingly smaller footprint, reducing their attack surface [3]. However, the more trust/privilege is assigned to hypervisors, the greater the motivation for an attacker to come up with possible ways to subvert their operation.

Hardware extensions for virtualization will play an important role in helping secure implementations, as several virtualization flaws exist due to their nature of being a software-based solution [48].

As with all computing technologies, virtualization presents its own security risks. Some of these issues inherently arise due to the nature of technology, while many occur when virtualization technology is deployed incorrectly. Too often, IT professionals make the mistake of relying solely on backups, firewalls, password and security tools to secure their data centers, but this approach does not cover all the bases [49].

This study introduced the the security considerations, concerns, and implications associated with virtualized environments. It mentioned various virtualization approaches and presented a series of security threats in a virtualized environment. It is important to consider the security threats that come with virtualization technology to have an efficient and effective infrastructure installed, as well as applying suitable defense mechanisms.

This document is an exhaustive overview of security in virtualized environments in recent years. The main objective was to help security professionals and IT professionals who are responsible for infrastructure virtualization, since virtualization has effectively changed the way we look at and treat IT. Despite all the benefits of virtualization, it also comes with a set of security risks. Virtualized assets are more difficult to protect than physical servers and require specialized tools and training to be managed [49].

The key to creating a truly effective cybersecurity strategy is to take a multilayered approach to securing both VMs and the virtualization stack [49].

In the near future, we would like to conduct an extensive study on approaches to mitigate security threats in cloud environment.

9. REFERENCES

[1] J. S. Reuben, “A Survey on Virtual Machine Security,” 2007.

[2] D. Rosado, R. Gómez, D. Mellado, and E. Fernández-Medina, “Security Analysis in the Migration to Cloud Environments,” Futur. Internet, vol. 4, pp. 469–487, 2012.

[3] M. Pearce, S. Zeadally, and R. Hunt, “Virtualization: Issues, security threats, and solutions,” ACM Comput. Surv., vol. 45, pp. 17:1-17:39, 2013.

[4] T. Ormandy, “An Empirical Study into the Security Exposure to Hosts of Hostile Virtualized Environments,” in CanSecWest 2007, 2007, pp. 1–10.

[5] N. Aaraj, A. Raghunathan, and N. K. Jha, “Virtualization-assisted Framework for Prevention of Software Vulnerability Based Security Attacks,” 2007.

[6] A. Azab, P. Ning, Z. Wang, X. Jiang, X. Zhang, and N. Skalsky, “HyperSentry: Enabling Stealthy In-context Measurement of Hypervisor Integrity,” 2010, pp. 38–49.

[7] F. Bazargan, C. Yeun, and J. Zemerly, “State-of-the-Art of Virtualization, its Security Threats and Deployment Models,” Int. J. Inf. Secur. Res., vol. 3, 2013.

[8] D. Tank, A. Aggarwal, and N. Chaubey, “Virtualization vulnerabilities, security issues, and solutions: a critical study and comparison,” Int. J. Inf. Technol., 2019.

[9] VMware, “Achieving Compliance in a Virtualized Environment,” 2008.

[10] M. Cobb, “A preview of PCI virtualization specifications,” 2011.

[11] CVE Details, “Vmware Esxi : CVE security vulnerabilities, versions and detailed reports.” [Online]. Available: https://www.cvedetails.com/product/22134/Vmware-Esxi.html?vendor_id=252.

[12] S. Jagathpal, “Information Security Blog,” 18-Feb-2010. [Online]. Available: http://shobhajagathpal.blogspot.com/2010_02_01_archive.html.

[13] S. Orrin and O’Berry David, “Building Security Beneath the OS - The Security Content Automation,” 2011.

[14] K. Kortchinsky, “Cloudburst: Hacking 3D (and Breaking Out of VMware) for Black Hat USA 2009,” 2009.

[15] A. Pingios, “CVE-2009-3692: VirtualBox VBoxNetAdpCtl Privilege Escalation,” 2009. [Online]. Available: https://xorl.wordpress.com/2009/10/13/cve-2009-3692-virtualbox-vboxnetadpctl-privilege-escalation/.

[16] D. D. Zovi, “Hardware virtualization based rootkits.” Black Hat USA, 2006.

[17] M. P. Souppaya, K. Scarfone, and P. Hoffman, “Guide to Security for Full Virtualization Technologies,” 2011.

[18] C. Li, A. Raghunathan, and N. K. Jha, “Secure Virtual Machine Execution under an Untrusted Management OS,” in Proceedings - 2010 IEEE 3rd International Conference on Cloud Computing, CLOUD 2010, 2010, pp. 172–179.

[19] A. Baruchi and R. L. Piantola, “Análise Quantitativa de Técnicas de Virtualização Como Ambiente de Testes.”

[20] R. Morabito, J. Kjällman, and M. Komu, “Hypervisors vs. Lightweight Virtualization: A Performance Comparison,” 2015.

[21] K. Adams and O. Agesen, “A Comparison of Software and Hardware Techniques for X86 Virtualization,” SIGOPS Oper. Syst. Rev., vol. 40, no. 5, pp. 2–13, Oct. 2006.

[22] VMware, “VMware Understanding Full Virtualization, Paravirtualization, and Hardware Assist,” 2008.

[23] R. P. Goldberg, “Survey of virtual machine research,” Computer (Long. Beach. Calif)., vol. 7, no. 6, pp. 34–45, Jun. 1974.

[24] N. Kiyanclar, “A Survey of Virtualization Techniques Focusing on Secure On-Demand Cluster Computing,” 2005.

[25] TechNavio, “Global Endpoint Server Security Market 2011-2015.”

[26] L. Wood, “Research and Markets: Global Endpoint Server Security Market 2011-2015 | Business Wire,” 13-Aug-2012. [Online]. Available:

https://www.businesswire.com/news/home/20120813005608/en/Research-Markets-Global-Endpoint-Server-Security-Market.

[27] G. J. Popek and R. P. Goldberg, “Formal Requirements for Virtualizable Third Generation Architectures,” Commun. ACM, vol. 17, no. 7, pp. 412–421, Jul. 1974.

[28] S. Ray, “Towards a Formalization of the X86 Instruction Set Architecture,” 2008.

[29] F. Tsifountidis, “Virtualization Security: Virtual Machine Monitoring and Introspection,” 2011.

[30] C. Strachey, “Time sharing in large, fast computers.,” in IFIP Congress, 1959, pp. 336–341.

[31] J. McCarthy, “Reminiscences on the History of Time-Sharing,” IEEE Ann. Hist. Comput., vol. 14, no. 1, pp. 19–24, Jan. 1992.

[32] J. Howlett, “The Atlas Computer Laboratory,” IEEE Ann. Hist. Comput., vol. 21, no. 1, pp. 17–23, Jan. 1999.

[33] D. Morris, F. H. Sumner, and M. T. Wyld, “An Appraisal of the Atlas Supervisor,” in Proceedings of the 1967 22nd National Conference, 1967, pp. 67–75.

[34] B. S. Brawn, F. G. Gustavson, and E. S. Mankin, “Sorting in a paging environment,” Commun. ACM, vol. 13, pp. 483–494, 1970.

[35] P. J. Denning, “Performance Evaluation: Experimental Computer Science at its Best,” 1981.

[36] J. Hoopes, Ed., “Chapter 1 - An Introduction to Virtualization,” in Virtualization for Security, Boston: Syngress, 2009, pp. 1–43.

[37] S. E. Madnick and J. J. Donovan, “Application and Analysis of the Virtual Machine Approach to Information System Security and Isolation,” in Proceedings of the Workshop on Virtual Computer Systems, 1973, pp. 210–224.

[38] J. C. C. dos Santos Ramos, “Security challenges with virtualization,” Universidade de Lisboa, 2009.

[39] V. Bourne, “Unleashing the Power of Virtualization,” 2010.

[40] P. Barham et al., “Xen and the Art of Virtualization,” SIGOPS Oper. Syst. Rev., vol. 37, no. 5, pp. 164–177, Oct. 2003.

[41] A. Whitaker, M. Shaw, and S. D. Gribble, “Scale and Performance in the Denali Isolation Kernel,” SIGOPS Oper. Syst. Rev., vol. 36, no. SI, pp. 195–209, Dec. 2003.

[42] M. D. Schroeder and J. H. Saltzer, “A Hardware Architecture for Implementing Protection Rings,” Commun. ACM, vol. 15, no. 3, pp. 157–170, Mar. 1972.

[43] J. Franklin, M. Luk, J. M. McCune, A. Seshadri, A. Perrig, and L. van Doorn, “Remote Detection of Virtual Machine Monitors with Fuzzy Benchmarking,” SIGOPS Oper. Syst. Rev., vol. 42, no. 3, pp. 83–92, Apr. 2008.

[44] P. Ferrie, “Attacks on Virtual Machine Emulators,” 2007.

[45] T. Liston and E. Skoudis, “On the Cutting Edge: Thwarting Virtual Machine Detection.”

[46] T. Garfinkel, K. Adams, A. Warfield, and J. Franklin, “Compatibility Is Not Transparency: VMM Detection Myths and Realities.,” 2007.

[47] N. M. Upadhyay and R. S. Singh, “An effective scheme for memory congestion reduction in multi-core environment,” J. King Saud Univ. - Comput. Inf. Sci., 2020, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1319157820303888.

[48] G. Pék, L. Buttyán, and B. Bencsáth, “A Survey of Security Issues in Hardware Virtualization,” ACM Comput. Surv., vol. 45, no. 3, Jul. 2013.

[49] M. Comeau, “Protect your infrastructure with virtualization security management,” 2017. https://searchservervirtualization.techtarget.com/tip/Protect-your-infrastructure-with-virtualization-security-management.