Security aspects of virtualization

FEBRUARY 2017

About ENISA

The European Union Agency for Network and Information Security (ENISA) is a centre of network and information security expertise for the EU, its member states, the private sector and Europe's citizens. ENISA works with these groups to develop advice and recommendations on good practice in information security. It assists EU member states in implementing relevant EU legislation and works to improve the resilience of Europe's critical information infrastructure and networks. ENISA seeks to enhance existing expertise in EU member states by supporting the development of cross-border communities committed to improving network and information security throughout the EU. More information about ENISA and its work can be found at www.enisa.europa.eu.

Contact

For contacting the authors please use opsec@enisa.europa.eu
For media enquiries about this paper, please use press@enisa.europa.eu.

Acknowledgements

The analysis in this document was produced in collaboration with Antonio Maña (University of Málaga), Eduardo Jacob (Basque Country University), Lorenzo Di Gregorio (Intel Deutschland GmbH) and Michele Bezzi (SAP).

ISBN 978-92-9204-211-0, DOI 10.2824/955316

Executive Summary

Experience has shown that virtualization can provide a dramatic increase in the efficiency and effectiveness of complex organizations and communities, and is expected to constitute an important technological pillar of a thriving data-driven economy and the European single digital market. However, virtualization also bears a number of (new) security risks. First, some risks are shared with traditional computing environments and include, for instance, issues affecting operating systems, communication protocols, and applications. Second, the above issues may even be exacerbated by the use of virtualized components, producing a greater security impact. For instance, privilege escalation may have increased impact if the target of the escalation is the OS of the physical machine hosting a virtualized system. Finally, virtualization also introduces a number of virtualization-specific security issues that require ad hoc solutions. For instance, new security issues are related to multi-tenancy allowing cross-platform information flow between customers sharing the same physical host, and allowing adversaries to execute arbitrary out-of-the-guest code without owning the required access rights. Nowadays, as the basis of distributed infrastructures like the cloud, virtualized environments are adopted pervasively and therefore increasingly targeted by cyber-attacks. Even more elaborated and specialized attacks are currently being devised to exploit vulnerabilities or weaknesses at the virtualization layer.

This report provides an analysis of the status of virtualization security. In it, we present current efforts, emerging best practices and known security gaps, discussing the impact the latter have on environments based on virtualization technologies.

Objectives and content of the report

This report shall provide an overview of the status of security of virtualized environments. It gives the basis to understand issues and challenges related to virtualization security, as well as a discussion on common best practices for security protection in virtualized environments and gaps that need to be filled in to implement a secure virtualized environment.

This report targets, on one side, system developers/administrators providing technical insights for securing their systems and, on the other side, policy makers and regulators providing an overview of relevant threats and weaknesses, and related countermeasures. It provides a better understanding of the opportunities, challenges and limits of virtualized systems and will improve the effectiveness of future policies and regulations.

Key findings

Virtualization systems and technologies have revolutionized the traditional view of ICT and are permeating many ICT domains and fields of today's society. For instance, virtualization is at the basis of server and desktop infrastructures, cloud computing, networking, and containerization. Virtualization supports better performance, greater transparency, and portability and interoperability by combining hardware resources, software resources, and network functionality into a single, software-based administrative entity. However, the price we pay for such advantages is a negative effect on security properties of systems, thus calling for ad hoc management solutions. In fact, by introducing a complex mix of software components, each with its own administrator privileges, virtualization technologies enlarge the IT system attack surface, with an increase of security risks. Virtualized systems introduce important security gaps that need to be taken into account when deploying strong and secure virtualized infrastructure. Among them, performance and scalability of security solutions must not interfere with security protection, the multi-

tenant nature of virtualized systems must be appropriately managed to provide strong isolation between tenants operations, assurance solutions must be supported to evaluate the behaviour of virtualized infrastructure and support a posteriori forensics analysis, privacy and data protection should be further strengthened in environments where multiple entities are operating on the same infrastructure.

List of recommendations

A set of recommendations for next-generation countermeasures emerge from the report. A brief overview of these recommendations is provided in the following; for a detailed discussion please refer to Section 5.2.

Recommendations for policy makers and data owners

- Policy makers and regulators should define clear roadmaps and security guidance requirements ruling virtualization system deployment and management.
- Clear and ad hoc standards need to be defined to accomplish the nature of virtualized systems.

Recommendations for developers of virtual infrastructures and administrators

- System developers/administrators need to depart from the approaches used in traditional physical environments, focusing on the new technological layers that are added in virtual systems.
- Administrators should clearly identify virtualized components used in their environment, so that the selection of solutions mitigating risks and threats will be simplified.
- There is a need of looking for specific solutions, identifying successful new security products and staying focused on any updates.

Organizational and Human Resources Recommendations

- Training of all human resources involved in the process of managing virtualized environments, from specialized professionals to managers and users, is of paramount importance to reduce risks and impact of attacks.
- Assurance solutions should be integrated to monitor the correct behaviour of virtualized systems, on one side, and security solutions, on the other side, and to take corrective actions in case of misbehaviours.
- Service-level agreement definition and enforcement should consider the inherent multi-tenant nature of virtualized environments.

List of Acronyms

AC: Access Control

API: Application Programming Interface

ARP: Address Resolution Protocol

CEN: European Committee for Standardization

CIFS: Common Internet File System

NFS: Network File System

CMDB: Configuration Management Database

COBIT Control Objectives for Information and Related Technologies

CSCC: Cloud Standards Customer Council

CSP: Communication Service Provider

CVE: Common Vulnerability Enumeration

CVSS: Common Vulnerability Scoring System

CWE: Common Weakness Enumeration

DAS: Direct Attached Storage

DdoS: Distributed Denial Of Service

DNS: Domain Name Server

DoS: Denial Of Service

ETSI: European Telecommunications Standards Institute

EU: European Union

FDC: FLoppy Disk Controller

GDPR: General Data Protection Regulation

IaaS: Infrastructure As A Service

ICT: Information and Communications Technology

ISACA Information Systems Audit and Control Association

ISO: International Organization for Standardization

LDAP: Lightweight Directory Access Protocol

MMU: Memory Management Unit

NAP: Network Access Point

NAS: Network Attached Storage

NIST: National Institute of Standards and Technology

NVD: National Vulnerability Database

OS: Operating System

PaaS: Platform As A Service

PICSE: Procurement Innovation for Cloud Services in Europe

PRNGs: Pseudo-Random Number Generators

RACS: Recommendations for Assurance in the Cloud

RAM: Random Access Memory

REST: Representational State Transfer

SaaS: Software As A Service

SAN: Storage Area Network

SDKs: Software Development Kit

SDN: Software-Defined Networking

SLA: Service Level Agreement

SMM: System Management Mode

SNMP: Simple Network Management Protocol

SQL: Structured Query Language

SV: Server Virtualization

VDI: Virtual Desktop Infrastructure

VENOM Virtualized Environment Neglected Operations Manipulation

VM: Virtual Machine

VMM: Virtual Machine Monitor

XSS: Cross-site scripting

ISC²: System Security Certification Consortium

CSA: Cloud security Alliance

CA: Computer Associates

ONF: Open Network Foundation

TC: Trusted Computing

vTPM: virtual Trusted Platform Module

CP-ABE: Chipertext-Policy Attribute-Based Encryption

SETS: Systems and Emerging Technologies Security Research

PM: Policy Machine

Table of Contents

1. Virtualization Technologies and Environments

This section provides a short review of virtualization concepts, technologies and environments. We start with a brief history of the evolution of virtualization. We then discuss the main virtualization technologies, summarizing types and characteristics of virtualization technologies, and their components. Finally we identify a list of application scenarios for virtualization with substantial impact on the workings of current ICT infrastructures. These application scenarios are meant to give the broadest possible view of relevant applications of virtualization technology, though they are not exhaustive.

1.1 History of virtualization

The recent and widespread adoption of virtualization technologies has changed the traditional view of ICT. Virtualization refers to the set of activities aimed at creating a virtual version of real components, including computer-hardware platforms, operating systems, storage, and networking. In the general understanding, virtualization encompasses all those technologies needed to set up virtual machines that provide virtual resources or devices. Virtualized resources or devices have the same functionalities and external APIs as physical ones, but with different characteristics (e.g., performance, costs).

The virtualization concept is related to the concepts of emulation and simulation, which, while similar to virtualization, implement different approaches or paradigms. Emulation is an approach through which a system is executed as if it were another system. OSs, APIs, and operations are executed (emulated) on a machine for which they were not developed. The emulator replicates the exact behaviour of a piece of physical hardware, executing a copy of software by emulating the hardware for which the software was developed. Simulation, on the other hand, simulates the behaviour of a given system. It aims to achieve the same result as an emulator, but requires rewriting part of the program to be simulated. Virtualization provides techniques for using resources and devices without considering their position and physical layout. It supports an encapsulated environment, guaranteeing machines isolation, hardware independence and hardware partitioning. Generally speaking, emulation and virtualization represent a target system accurately, but at high costs, whereas simulation is cheaper and more flexible, but less accurate. Table 1-1 summarizes the main characteristics of simulation, emulation, and virtualization.

Table 1-1

#	CONCEPT	MAIN CHARACTERISTICS
#1	Simulation	Approximate the behaviour of the real system, requires rewriting software, cheap and flexible, loses accuracy
#2	Emulation	Emulate the behaviour of the real system, executes unmodified code, accurate and flexible, expensive
#3	Virtualization	Virtualize the exact behaviour of a real component, cross-platform, accurate and flexible, expensive

Although the virtualization “revolution” is quite recent, having begun at the end of 1990s, the virtualization concept dates back to the 1960s with the development of virtualization techniques on mainframes that aimed to provide concurrent execution of processes and applications, thus solving the problem of resource

underutilization. The first virtualized system was the IBM S/360 Model 67 mainframe released in the 1960s, which virtualized all hardware interfaces through what was called the *Virtual Machine Monitor* (VMM).[1] Following this effort, the term hypervisor as the layer supporting the execution of operating systems on other operating systems was introduced in the 1970s and the concept of logical partitioning in the 1980s. At the time, in 1974, Popek and Goldberg published the paper titled "Formal requirements for virtualizable third generation architectures". [2] The paper represents a mainstay of virtualization techniques, providing guidelines for designing virtualized computer architectures that are still valid for determining whether a given architecture will support efficient virtualization. The guidelines defined a formal model for third generation computer systems, which was used to derive sufficient and necessary conditions to establish whether a given architecture can support a VMM. In other words, they tried to answer the question: "Can this hardware architecture support virtualization efficiently"? The approach proposed by Popek and Goldberg is based on the concept of the VMM as any control program that supports efficiency, resource control, and equivalence properties, as well as on a classification of instructions as privileged, control sensitive, and behaviour sensitive. It consists of two main theorems that define conditions for evaluating a given architecture's support of virtualization and its readiness for implementing recursive virtualization.

After a couple decades (the 1980s and 1990s) when the application of virtualization was limited by the client-server paradigm, in the last 20 years the 1960s problem of resource underutilization has revived to overwhelm ICT infrastructure. Many physical servers were deployed, leading to increased costs, problems due to failures and hardware obsolescence, and low flexibility in distributed systems. Furthermore, the need for many physical services increased infrastructure requirements in terms of maintenance, leases, networking, floor space, cooling, power, and disaster recovery. Major players in the ICT industry therefore started their own (opensource) virtualization strategy, bringing virtualization to the fore. Virtualization strategies evolved from the idea of simply running one system on another into the idea of a tool for maximizing the use of resources. Recently they have further evolved into a potential model for offering computing platforms as services (see section 1.4 for greater detail). According to a Gartner study in July 2015,[3] 75% of x86-server workloads are virtualized. They are increasingly lightweight, support more workloads, and allow for agile development. This scenario gives the idea of the magnitude of the importance virtualization is acquiring in everyday digital life. This is further boosted by the growth of new paradigms, such as cloud and software-defined networking. Focusing on the client side of virtualization, an IDC study in 2015[4] presented an assessment of virtual client-computing software by analyzing the capability and business strategy of major vendors. This study shows that client virtualization has entered maturity.

In general, virtualization solutions allow different users to manage and share physical hardware by supporting multiple shared environments that are isolated, while running on the same infrastructure. Virtualization introduces many benefits that strengthen ICT flexibility and efficiency, summarized as follows:

- Server consolidation and reduced costs for system operation and management (e.g., restoring servers after hardware failure), while keeping needed computing power.

---

[1] Shannon Meier, IBM Systems Virtualization: Servers, Storage, and Software, April 2008, http://www.redbooks.ibm.com/redpapers/pdfs/redp4396.pdf

[2] G.J. Popek, R.P. Goldberg. "Formal requirements for virtualizable third generation architectures". Communications of the ACM 17 (7): 412–421, July 1974

[3] T.J. Bittman, P. Dawson, M. Warrilow, Magic Quadrant for x86 Server Virtualization Infrastructure, July 2015 http://www.gartner.com/technology/reprints.do?id=1-2JGMVZX&ct=150715&st=sb

[4] R. Young and D. Laing, IDC MarketScape: Worldwide Virtual Client Computing Software 2015 Vendor Assessment, June 2015

- Optimized resource utilization, responding to application and user requirements dynamically at runtime, allowing users to share resources.
- Multiple execution environments, where users can select the environment that best suits their requirements in a given timeframe.
- Simplified management through a single view of all (distributed) resources, supporting interoperability between heterogeneous systems and centralized control of the environment.

To sum up, virtualization is permeating many fields of today's society and is widely adopted by ICT providers. Virtualized systems provide better performance, greater transparency, and portability and interoperability by combining hardware resources, software resources, and network functionality into a single, software-based administrative entity. They promise increased support for green IT. Despite these undebatable advantages, virtualization may also negatively affect non-functional properties of systems, such as security and efficiency, thus calling for ad hoc management solutions. This document focuses on security issues and provides a review on the security status of virtualized systems.

1.2 Virtualization components

Virtualization techniques and virtualized architectures introduce an additional layer of execution, including their own administrator role (virtualization admin), which require proper management and security protection. This layer is made up of several different components, each with a role in the virtualization process, each representing a potential new target for malicious attacks. This section discusses some of these components, below, serving for the remainder of the report as a basis for describing virtualization threats, vulnerabilities, and risks. [5,6]

- **The hypervisor** is the component that acts as a mediator between virtual machines and the underlying physical devices. It mediates all hardware requests by the virtual machines down to the physical hardware, sharing physical devices as resources. It implements the virtual machine monitor providing virtualized hardware (hardware abstraction) to virtual machines. It can be of two types, bare-metal or hosted, as discussed in section 1.3. Examples of hypervisors are VMware ESXi, Xen Hypervisor, VirtualBox, and Microsoft Hyper-V, to name but a few.
- **The virtual machine monitor** is an application component of the hypervisor that keeps track of activities carried out by virtual machines (i.e., it manages VM applications), forwards hardware request to physical resources, provides replicated platforms, and supports resource sharing between different virtual machines. It has the responsibility to guarantee end users virtualization transparency.
- **Guest machines**, also known as virtual machines, instantiate the virtualized (encapsulated) system made of the operating system and applications, using the hardware abstraction provided by the virtual machine monitor. 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 operating system to support virtualization heterogeneity. A special, guest-machine case is the container (see Section 1.4.5), an operating-system-level virtualization in which the kernel of an operating system allows the existence of multiple isolated user-space instances.

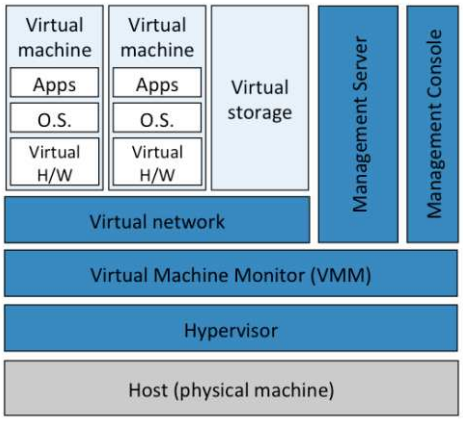
[5] G. Pek, L. Buttyan, B. Bencsath, "A survey of security issues in hardware virtualization," ACM Comput. Surv. 45, 3, Article 40, June 2013.

[6] Bryan Williams, Virtualization System Security, IBM Corporation, 2010

- **The host machine** is the real physical machine and its operating system (host operating system) that hosts the virtualized environment. The host operating system directly manages the physical hardware underlying the virtualized environment and is where the hypervisor runs. We note that sometimes the term "host operating system" 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). When necessary, we will refer to them as privileged VMs or admin VMs.
- **The management server** is the virtualization platform made up of a set of components for directly managing the virtual machines, consolidating services, allocating resources, migrating virtual machines, and assuring high availability, to name but a few. Examples of management servers include VMware vSphere and XEN XenCenter.
- **The management console** is the component that provides access to a management interface to the virtualization product for configuring and managing virtual machines. Virtual machines can thus be added, modified, deleted or configured. The management console can be provided as a standalone client or via a web interface to visually handle management server functionalities. Examples of management consoles include the VMware vSphere client console and the VMware vSphere web client.
- **The network components** that facilitate the development of virtual networks, where virtual network devices (e.g., switches, routers) are completely controlled though software and the network protocols and stack are simulated to replicate physical ones insofar as possible. Virtual machines are connected the same way as physical machines and built on host-machine physical network infrastructure to connect to the public network. An example of a reference implementation of a completely software-based, virtualized network component is Open vSwitch (see section 1.4). This product represents full layer2-layer3 virtual network equipment with support for the OpenFlow SDN protocol. The distributed network topology is kept coherent across physical devices by a software controller such as OpenDaylight.
- **Virtualized storage** that provides all the components for abstracting physical storage in a single storage device that can be accessed either over the network or through a direct connection. Storage virtualization introduces additional management overhead, due to the fact that 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 technology, including direct attached storage (DAS), storage area network (SAN), and network attached storage (NAS). Examples of these devices include the RAID arrays hosted inside a server computer (DAS), the storage device collecting all datacenter data such as the EMC VNX7500 (SAN), and the simple storage component that offers network file-level access through a wide variety of application protocols like CIFS or NFS.

Figure 1-1 shows a graphical representation of the components of virtualization.

Figure 1-1 Components of virtualization



The virtualization layer (blue color in the Figure 1-1) introduces a complex mix of software components at different levels of the computing architecture (e.g., operating system, communication, management, interface), each with its own administrator privileges, thus enlarging the attack surface. Virtualization techniques, while providing undebatable advantages to security, also come with increased security risks that must be taken into account when deploying strong and secure virtualized infrastructure.

1.3 Virtualization technology classification

Virtualization technologies are initially classified according to their degree of hardware emulation. We can distinguish between approaches that provide *full hardware emulation* and approaches that provide *hardware virtualization* (or *OS virtualization* or *partial hardware emulation*) as follows.[7,8]

Full hardware emulation allows executing an unmodified system (guest OS) in a different host architecture. It emulates all features of a software system or device on a hardware platform with a different instruction set. Examples of solutions that support full hardware emulation include Bochs, QEMU, VirtualPC.

Hardware virtualization defines a class of virtualization technologies in which a software system or device is executed on a hardware platform with the same instruction set. We note that there is not always a strict separation between hardware virtualization and emulation, since in some cases hardware emulators can be used for device virtualization. Hardware virtualization is further classified in three classes as follows.

- **Full virtualization** supports the virtualization of (x86) systems by simulating the underlying hardware. The hardware is simulated in software by each virtual machine. The guest OS is completely separated from the underlying hardware, access to which is mediated by the virtualization layer (virtual machine monitor). The guest OS runs unmodified with no need for hardware or operating-system support. Full virtualization can be based on a mix of binary translation of kernel code and direct execution of user-level code. 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

[7] G. Pek, L. Buttyan, B. Bencsath, "A survey of security issues in hardware virtualization," ACM Comput. Surv. 45, 3, Article 40, June 2013

[8] VMware, Understanding Full Virtualization, Paravirtualization, and Hardware Assist, Whitepaper, https://www.vmware.com/files/pdf/VMware_paravirtualization.pdf

overhead. Examples of solutions supporting hardware virtualization include VirtualBox, Virtual PC, VMware, Win4Lin, Xen, and User Mode Linux.

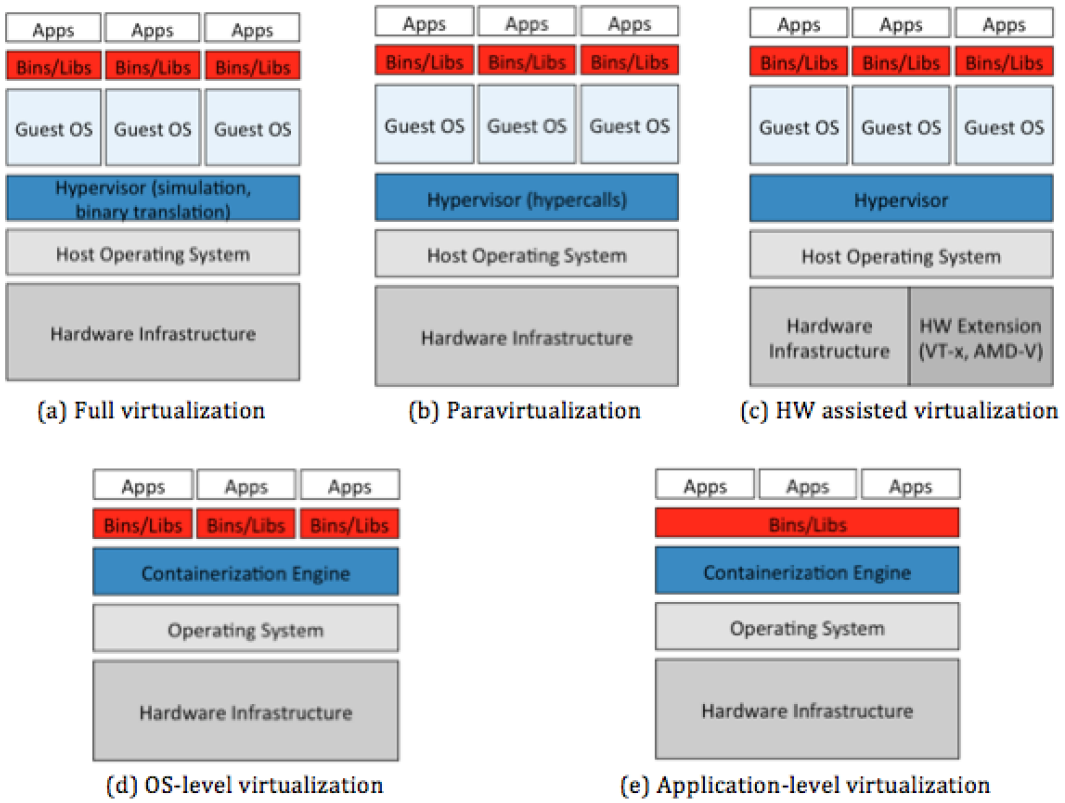
- **Paravirtualization** provides a lightweight virtualization technique where the hypervisor exposes hypercalls that can be directly called by a modified guest OS to simulate privileged instructions that are difficult to virtualize. The hypercalls implement a virtualized version of system calls and 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 (it does not require emulation of system resources), at the price of requiring changes to the guest operating system. Examples of solutions supporting paravirtualization include Xen, KVM/QEMU, and Win4Lin 9x.
- **Hardware-assisted virtualization** builds on hardware vendors' efforts to provide new features to support virtualization techniques. Intel Virtualization Technology (VT-x) and AMD's AMD-V were introduced midway through the first decade of the 2000s and provide a new execution mode that allows virtual machine monitors to run in a new privileged mode. It makes hardware extension available to the guest OS, providing better performance and reducing changes required by paravirtualization. Examples of solutions supporting hardware virtualization include VmwareWorkstation (64-bit), VirtualBox, Xen, KVM/QEMU, Parallels, and Microsoft Hyper-V.

In addition to the degree of hardware emulation, virtualization technologies can be classified on the basis of virtualization level. Above, we described 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. We can add two further classes:

- Operating-system-level virtualization is based on an operating system that supports multiple instances of isolated user-space, called containers. Each container can target a single application and install only the needed software and libraries to run this application. The host machine's hardware resources are partitioned between different guest machines. The host OS deploys many instances of guest OSes, with a lightweight execution of the operating system or application. Resources are assigned to containers that represent a set of processes, files, and partitions. This approach provides high performance, low overhead, and permits the execution of the same OS as the host machine. Examples of solutions supporting operating system-level virtualization include Docker, Virtuozzo, OpenVZ, and Solaris Containers.
- Application-level virtualization increases programs' portability between 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.

Figure 1-2 shows the high-level architecture of the main classes of virtualization.

Figure 1-2 Virtualization taxonomy



All the above virtualization technologies can be also classified according to the hosting type provided by the hypervisor. Two hosting types are possible: *native* (bare-metal virtualization or type I hypervisor) and *non-native* (hosted virtualization or type II hypervisor) virtualization. *Bare-metal virtualization* involves a virtualization layer directly installed on the host hardware that mediates requests from virtual machines. The hypervisor runs directly on the host hardware and communicates with physical devices. When bare-metal virtualization is considered, the host operating system and hypervisor are integrated in a single layer within the hypervisor. *Hosted virtualization*, on the other hand, involves a layered structure made up of the host hardware and host operating system (host machine), the hypervisor (running in the host operating system as a traditional computer program), and the guest operating system installed in a virtual machine. The hypervisor is therefore installed on top of the host operating system and coexists with applications already installed in the host operating system itself.

It is important to note that virtualization technologies must adapt to the surrounding environment made up of hardware and software technology and build on or interact with them to provide a secure, robust, and effective virtualized system. For instance, virtualization technologies need to adapt to the host machine's CPU architecture, as well as to the host operating system, if any. As a further example, X86 architectures have a system management mode (SMM) executed in Ring-2 with firmware for system-wide functions (mainly power management), reached through a dedicated "system management interrupt." Access to the latter functionality provides high privileges and must be monitored to secure not only the virtualized system but also the physical system. On the other hand, virtualized systems often rely on traditional software systems (e.g., operating systems) and need to consider their specificities to guarantee appropriate non-functional properties (e.g., security, performance). For instance, the security of a virtualized system depends

on the security of the guest operating systems, including the protection against all those attacks that are not peculiar to virtualized OSes but can target a generic OS installed on certain physical hardware.

Finally, an orthogonal class of virtualization techniques or products can be defined to embrace the concept of *network virtualization*, where entire distributed networks are virtualized including devices and cables. Even the protocols are adapted to work within virtualized networks, providing a flexible, dynamic, and robust virtualization approach to networking. Within this concept there is an entire ecosystem of custom software products developed to keep the virtualized network under control, providing standard management-protocol support, like SNMP. Bearing in mind that every component in the datacenter network consists of a virtual instance, every product in this family is built from the ground up to offer an extreme flexibility through one or more SDKs or APIs and unleashes the full feature set available in software solutions. As a complementary aspect of network virtualization, the virtualization of network functions aims to optimize network services by running network functions (e.g., DNS, firewall, NAT, intrusion-detection system) in software, thus separating them from proprietary hardware appliances.[9]

Today various virtualization products have been developed by different players. Table 1-2 gives an overview of major products.

Table 1-2 Virtualization products

PRODUCT	CLASS OF VIRTUALIZATION	MAIN CHARACTERISTICS
VMware ESXi	Bare-metal, full virtualization	It is a full virtualization hypervisor originally called ESX Server. The system was initially released with an integrated management console. VMware ESX, later ESXi, is released as an installation image that provides binary implementation of the hypervisor plus a certified set of device drivers to access storage and networking hardware supported directly by the vendor itself.
XEN	Bare-metal, paravirtualization	The XEN hypervisor provides a paravirtualized virtualization environment. It is a bare-metal hypervisor begun as a research project at the University of Cambridge Computer Laboratory, now offered commercially by Citrix. XEN is distributed as source code that anyone can compile on a standard Linux host. After installation, the hypervisor is in charge of managing the boot sequence of the host computer and then boots the privileged domain kernel. The paravirtual model needs a slightly modified virtual-machine OS kernel to correctly run virtual machines as unprivileged domains on XEN.

[9] B. Han, V. Gopalakrishnan, L. Ji and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," in *IEEE Communications Magazine*, vol. 53, no. 2, pp. 90-97, February 2015.

Microsoft Hyper-V	Bare-metal, paravirtualization	Hyper-V is the reference hypervisor in the Microsoft Windows product family. As a paravirtualized, bare-metal hypervisor, Hyper-V has a privileged virtual machine containing device drivers to access hardware devices on the host computer. This particular domain is a Windows Server instance with the hypervisor role assigned.
Oracle VirtualBox	Hosted	Oracle VirtualBox is an opensource-hosted hypervisor available for Windows, Mac OS X, and Linux. The hypervisor core is installed as a host-operating-system extension like any other driver or kernel extension. This product's management interface, installed in the host operating system's user space, offers the ability to control every aspect of the hardware configuration of a generic virtual-machine instance.
VMware Workstation	Hosted	VMware Workstation is a hosted hypervisor available for Windows and Linux. It allows the creation of fully virtualized x86 and x86-64 virtual machines with support for all kinds of guest operating systems. VMware workstation also has support for paravirtualized drivers, called VMwareTools, to enhance the user experience.
KVM-QEMU	Bare-metal, hardware-assisted	KVM-QEMU is a set of software that represents the default solution to virtualization in the Linux ecosystem. KVM is a hardware-assisted hypervisor that leverages the virtualization extensions offered by Intel, with VT-x technology, and by AMD, with AMD-V. KVM is a purely kernel-space hypervisor without any support for user-space interaction. To solve this problem the QEMU project was modified to correctly expose the hardware model to the guest operating systems inside any virtual machines created.
Qubes OS	Operating-system level, paravirtualization	Qubes OS is a free GNU/Linux general purpose distribution with a full desktop environment. The distribution's main characteristic is that it is based on Xen. The user interacts with the privileged domain using the normal desktop environment. At application startup, Qubes OS instantiates a new unprivileged domain and executes the application inside this new virtual machine. The aim of this modus operandi is to isolate the execution of every application, thus providing much greater security and reliability.

Linux Containers	Operating-system level	LXC, also known as "Linux Containers," is an operating-system-level virtualization that can run multiple isolated Linux instances, each isolated inside a logical container. Every container has its own set of resources managed by the Linux kernel through the control groups (cgroups), a kernel feature that assigns priority and applies fine-grain resource limitation without needing to run a new operating-system instance. Aside from the cgroup feature, there is another subsystem, identified as namespaces, that implements isolation between every container execution environment.
Docker	Operating-system level	Docker is the leading technology for operating-system-level virtualization in the Linux environment. This product heavily relies on the presence of the new Linux init system call and specifically needs system-nspawn. Docker offers a level of abstraction that permits generic container deployment, using cgroups and namaspaces, extending them if needed through a dedicated library called libcontainer.

1.4 Application scenarios for virtualization

Virtualization technologies are becoming the cornerstone of many ICT products and solutions. They are adopted in a variety of environments, touching many of our daily activities and different fields in our society. From these various environments, we selected a set of widespread application scenarios, where virtualization makes the difference. Obviously, virtualization brings advantages and limitations. On the positive side, there is the ability to quickly restore a server after hardware failure and the capacity to mix different hardware platforms and even OSes to conform to your computing hardware infrastructure,[10] eliminating dependencies between hardware and software components. On the downside, data security, server authentication, and efficiency may be negatively affected. In the scenarios below, we highlight both the advantages and the drawbacks.

1.4.1 Server virtualization (SV)

Server virtualization represents the first application of virtualization techniques in section 1.3. The goal of this scenario is to achieve consolidation by sharing hardware resources among different enterprise servers. Server virtualization substantially reduces enterprise costs by limiting the need for hardware, while maintaining the same quality of service within the datacenter by increasing system utilization and reducing datacenter operating costs. This paradigm's success stems from the selection of the proper hypervisor to support, among other things, dynamic provisioning, workload management or isolation, effective migration, and fast reconfiguration. Server virtualization also yields a better match between long-term requirements and resources, higher redundancy, and simpler management of availability and reliability properties (e.g., disaster recovery, fault tolerance). Another fundamental aspect of server virtualization links to all facets of datacenter management. Server virtualization provides mechanisms to support remote management of different servers through a single and consistent interface, often based on web technologies (e.g., REST

[10] We may mention Amazon as a provider of different hardware/software platforms.

services). This supports automatic monitoring techniques that aim to observe specific events while maximizing global datacenter uptime.

Uptime, and more in general availability, of the virtualized infrastructure are not the only non-functional requirements to be preserved for virtualized servers. Servers often maintain the assets of an enterprise and need to implement high security standards. The consolidation of many servers on a single hardware machine comes with increased security risks, because the virtualization layer becomes a single point of failure. Successful attacks on virtualization infrastructure might allow the intruder to gain control of many servers and their data in one fell swoop. Furthermore, virtualization-layer administrators may have access rights to servers that were inconceivable few years ago. This scenario points to the need for stronger security techniques that focus on protecting virtualized servers from attacks on the virtualization layer.

1.4.2 Virtual desktop infrastructure (VDI)

The virtualization of desktop clients is the natural evolution of the virtualization techniques originally laid out for server virtualization. The main goal of desktop-virtualization products is to reduce the amount of resources and computational power at client side, thus centralizing such resources in a single point. Consequently, in this scenario, the operating system is seen as a commodity, where default configurations can be applied and a set of policies replicated by using virtual-machine templates. This approach points to the adoption of systems for application provisioning, which have a single location and management interface, allowing us to manage the application lifecycle consistently and securely.

Two main VDI paradigms exists: *i)* central VDI hosting, where the desktop computer, including the operating system and applications, is executed on the server machine, and *ii)* local VDI hosting, where the entire desktop computer environment is first downloaded and then executed on the local machine. Both paradigms require a fast and reliable communication network. Indeed, lack of connectivity leads to a scenario in which clients are unable to use their own computers. On the other hand, when the network works properly, the client desktop is available anywhere, anytime.

As already noted for server virtualization, in addition to availability, security of VDI is paramount for its success and widespread adoption. But is VDI a means to increase desktop security or an opening for attackers? VDI can both contribute to strengthening the traditional desktop and introduce new threats or risks. Focusing on VDI security advantages, data is kept behind a firewall and protected from unauthorized access, virtual desktops reduce the risks introduced by using protocols like VPN to connect potentially infected devices to the corporate network, and VDI provides centralized system-image administration. The security disadvantages of VDI are closely linked to the peculiarities of a virtualized infrastructure. Desktop clients share the same set of resources, providing only logical partitioning and isolation. Furthermore, centralized management and distribution make the VDI infrastructure a single point of attack and require clients to exchange data, usually stored locally, over the public network.

1.4.3 Cloud computing

According to the NIST,[11] "*Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.*"

[11] P. Mell and T. Grance, The NIST Definition of Cloud Computing, September 2011 http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

The cloud-computing paradigm thus supports a new vision of IT whereby software applications and computational resources are released as services and used on a pay-as-you-go basis over virtualized ICT infrastructure accessible through the internet.[12] Cloud computing is becoming the preferred way to provide IT services. The cloud paradigm comes with several advantages for customers (i.e., end-users and service providers), which can outsource part of their business (that require great IT skill) to the cloud, thus reducing costs for owning, operating, and maintaining computing infrastructure, increasing flexibility, and benefiting from scalable infrastructure. Furthermore, cloud computing provides: *i)* rapid elasticity that allows resources to scale out and down depending on demand and gives end users stable quality of service and *ii)* metering capacity for controlling and optimizing resource usage.

Cloud computing provides three service models, as follows:

- **Software as a service** (SaaS) supports the component-based development model, developing a generic software as a composition of atomic, independent components. The SaaS model allows consumers to use and compose applications deployed on the cloud. Applications can be accessed remotely, for instance through a web browser or a program interface. This model's success is due to defining standard, self-descriptive interfaces that simplify the integration and reuse of single applications. The SaaS user does not control the physical infrastructure, operating system, network, storage, and application capabilities. The user merely manages certain specific application-configuration settings.
- **Platform as a service** (PaaS) targets developers and provides a fully dedicated environment for developing and deploying custom applications. PaaS solutions support different software stacks and application frameworks, including programming languages, libraries, services, and tools for application management. To provide a complete development framework, PaaS solutions are closely coupled with versioning systems and continuous integration. PaaS users do not control the physical infrastructure, operating systems, network, or storage, though they manage the whole application lifecycle and corresponding environment configuration.
- **Infrastructure as a service** (IaaS) provides all the components of a traditional datacenter, both computational resources and network connectivity. It offers processing, storage, networks, and generic computing resources on demand (e.g., encapsulated in a virtual machine) that can be freely managed by end users. The user does not control the physical infrastructure but manages the operating system and storage, and can install any application. An IaaS provider distributes resources on demand, for instance virtual machines with a given operating system and software stack.

The convenience introduced by cloud computing, in terms of flexibility and reduced costs for owning, operating, and maintaining computing infrastructure, comes at the price of increased security risks and concerns. Users deploying a service in the cloud lose full control over their data and applications, which are fully or partially in the hands of cloud providers. In addition, it makes the end user unaware of the infrastructure's performance and capacity constraints. On top of that, security represents one of the main problems hindering the shift of customers to the cloud. Security issues and requirements affect all layers of the cloud stack (service, platform and infrastructure layers). Though all important, this document focuses on IaaS security issues and challenges, where traditional virtualization functionalities are usually used. A breach in the security of the virtualized infrastructure opens the door to attacks on the platform and service layers, making such attacks disruptive.

---

[12] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," in Tech. Rep. UCB/EECS-2009-28, EECS Department, U.C. Berkeley, February 2009.

1.4.4 Software-defined networking (SDN)

Computer networks are traditionally implemented using dedicated hardware devices (e.g., switches, routers), which provide networking functionalities by means of a mix of open and proprietary protocols and interfaces. This scenario substantially reduces process automation and slows down service provisioning, reducing the capacity to react to evolving business needs. In the last decade, the network has evolved to a point where the need for elasticity and automation in configuring services at layers 2 and 3 of the ISO/OSI stack has grown. In response to such demand, the virtualization concept has been applied in the context of computer networks, leading to the establishment of software-defined networking.[13]

Software-defined networking aims to provide a flexible, agile way to manage networks, mimicking the approach already in use for managing virtualized servers and infrastructures. It provides a software approach to build and manage complex network topologies by abstracting higher-level functionalities. A network administrator is provided with a centralized console that can be used to configure the network and its services, without needing to communicate directly with the devices. The basis of this approach is strict separation between the control logic that decides whether to forward frames or packets (*control plane*), from devices that physically implement such forwarding (*data or forwarding plane*). Specifically, the control plane implements centralized forwarding policies for how frames or packets are distributed within the network. It provides a global overview of the network with a single management interface. The data (forwarding) plane represents the physical counterpart, where network devices become completely programmable, by implementing the data rules dictated by the administrator. On top of SDN, virtual devices are defined to implement computer networks in software. For instance, solutions as Open vSwitch or OpenContrail implement virtual, distributed, and multilayer switches for virtualized environments. Both can be used to implement devices within hypervisors or to manage physical devices acting as switches.

Communication between the data plane and the control plane requires an entire set of protocols, for which OpenFlow and OpenDaylight emerge as opensource proposals. These approaches support both physical and virtual devices. They also permit network components to communicate, thus assuring the network functions and is monitored properly.

To sum up, SDN architecture is programmable, flexible, and agile. It supports centralized management and can be based on open standards because it is vendor-neutral. It is complemented by network-function virtualization, where network services are optimized by separating network functions (e.g., DNS, firewall, NAT, intrusion detection system), run in software, from the underlying hardware. However, virtualized networks require rethinking network security. First, as in other scenarios, the security of the network is affected by the security of the virtualization layer and all its protocols. Second, distributed device management mandates strong requirements for communication security. Finally, because it is implemented via software, the configuration of virtual switches can easily be modified, for instance, to cause a denial of service, to slow down a given client's communication, or simply to give a client an advantage over its competitors.[14]

---

[13] B. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," Communications Surveys & Tutorials, IEEE 16, no. 3 (2014): 1617-1634.

[14] C.A. Ardagna, E. Damiani, "Network and Storage Latency Attacks to Online Trading Protocols in the Cloud," in Proc. of the International Conference on Cloud Computing, Trusted Computing and Secure Virtual Infrastructures, Amantea, Italy, October, 2014

1.4.5 Containerization

Originally, machine virtualization was bound to the creation of virtual instances of a complete system (virtual machines).[15] [16] The behavior of a complete system is fully replicated, from hardware to operating system, at the price of increasing the cost of virtualization. This results in a scenario whereby the management of the single virtual machine is independent and very similar to managing a physical machine. To address the need for lightweight and efficient virtualization of specialized applications, virtual instances have recently been created on the basis of the container concept. A container is basically a partition of the system's resources that resides in userland, which targets specific sets of applications. It is based on the concepts of file-system segregation and namespace. The main advantage of containerization is that it provides an approach for service provisioning with reduced overhead compared to the overhead of traditional approaches that need to maintain a complete operating-system instance. Each container is built on top of a single operating system and contains the minimum set of binaries or libraries for executing a given application or service.

As with the traditional virtualization approach, the containerization paradigm also points to a scenario where the resources of a single physical machine are shared and only logically partitioned. This architecture thus brings with it the need to carefully analyze the security aspects of containerization, especially attacks that might target the containerization engine.

We note that the hybridization of the above application scenarios is possible in practice. For instance, an approach to server virtualization can be deployed over a cloud infrastructure. However, for simplicity, but without loss of generality, they are considered separately below. In any event, virtualization-security analysis in hybrid application scenarios would clearly resemble the virtualization-security analysis of each component scenario.

[15] J. Turnbull, The docker book, 2016, https://www.dockerbook.com/

[16] M.J. Scheepers, Virtualization and Containerization of Application Infrastructure: A Comparison, 2014, http://referaat.cs.utwente.nl/conference/21/paper/7449/virtualization-andcontainerization-of-application-infrastructure-a-comparison.pdf

2. Assessment of Threats and Risks to Virtualized Environments

In this section, starting from the components identified in section 1, we provide an overview of virtualization threats, weaknesses, and vulnerabilities. We also present a risk evaluation and propose a prioritization of vulnerabilities.

2.1 Terminology

We adopt the following standard definitions:[17]

- **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.
- **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) within software, operations or infrastructure, in which the weakness can be used by a party to perform actions that were not specifically granted to the party who takes advantage of the weakness.
- **Impact:** the effect of an event, incident or occurrence. In cybersecurity, this means the effect of a loss of the 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 particular potential vulnerability, as well as the resulting impact of that adverse event on the organization.

A *threat* represents the potential for an attacker (*threat agent*) to exploit one or more system *weaknesses* through a concrete *vulnerability*. We note that vulnerabilities are system-specific, whereas threats and weaknesses are generic. We also note that a single vulnerability may exploit more than one weakness. An exploited vulnerability produces an *impact* on the system. The impact of the vulnerability, along with the likelihood of its being exploited, yields an estimate of the risk associated with the vulnerability itself.

In the remainder of this section, we first classify virtualization threats and threat agents. We then present our classification of the weaknesses affecting virtual environments and their corresponding vulnerabilities. Finally, we present a risk evaluation at both the weakness level and the vulnerability level, specifying a methodology for prioritizing vulnerabilities.

[17] All definitions come from ENISA glossary (see https://www.enisa.europa.eu/topics/threat-risk-management/risk-management/current-risk/risk-management-inventory/glossary), ENISA Threat Taxonomy v.1.0, January 2016 (see https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends/enisa-threat-landscape/etl2015/enisa-threat-taxonomy-a-tool-for-structuring-threat-information), and MITRE (see Common Weakness Enumeration in https://cwe.mitre.org).

2.2 Categorization of threats to virtualization

Virtualization shares with traditional computing environments a number of security issues that affect operating systems, communication protocols, and applications. In some cases, such issues are even exacerbated by the presence of virtualized components, requiring special care to address them. For example, traditional operating-system security issues, like the ones allowing privilege escalation, may produce greater impact if the target of the escalation is the OS of the physical machine hosting a virtualized system and its users. On the other hand, virtualization also introduces a number of virtualization-specific security issues that require ad hoc solutions. For instance, new security issues introduced by virtualization involve: i) multitenancy allowing cross-platform information flow between customers who share the same physical host, ii) adversaries able to execute arbitrary out-of-the-guest code without owning the required access rights, and iii) special security requirements in virtualized storage for keeping data secure during cloud events such as migration.

2.2.1 Threat taxonomy

To address the security concerns introduced by a virtual environment, a threat model is needed to classify all possible attacks. In the following, we present a general consequence-based categorization of threats as defined in IETF RFC 2828,[18] but specifically focusing on virtualization. We then also report on threats to the physical infrastructure peculiar to virtualization, while we choose not to dwell on generic threats such as physical attacks (deliberate and intentional), natural and environmental disasters, outages and failures or malfunctions (e.g., malfunction of the ICT supporting infrastructure). The impact of generic threats on virtualized environments is, indeed, often similar to their affect on any software, operations or infrastructure.

(Unauthorized) Disclosure: gaining unauthorized access to protected information

Sensitive information may be erroneously exposed or acquired by unauthorized entities, possibly circumventing security protections that are in place. Specifically, sensitive data may be *exposed* to unauthorized entities either erroneously (i.e., human errors), intentionally, or due to residue remaining in the systems (i.e., scavenging). Sensitive information can be the target of *interception*, for instance while data are in transit (e.g., eavesdropping or sniffing over a network). It may be indirectly *inferred*, for example by inspecting by-products of communication or indirect information flows. Or system protection may be circumvented via *intrusion*.

In virtual environments, where physical resources are shared between tenants, there may be a set of behaviours that result in the disclosure of sensitive information. For instance, *exposure* via scavenging in virtualized environments is even more serious[19] than in physical systems. While *interception* is a common threat in physical systems (e.g., networking environments), its effect is further exacerbated in virtual environments because it permits cross-inspection of various tenant's data flow, as well as topology inference that could serve to set up a denial-of-service attack. In virtualized environments, it is difficult to counteract intrusion threats, since intruders may obtain privileges through resources that are not the direct target of

[18] R. Shirey (May 2000) Network Working Group RFC 2828: Internet Security Glossary, in . The document provides glossary and it is intended to improve the comprehensibility of writing that deals with Internet security. It can be considered compatible with terminology used in other ENISA documents, such as the Threat Landscape reports.

[19] B. Albelooshi, K. Salah, T. Martin, E. Damiani, "Experimental Proof: Data Remanence in Cloud VMs", CLOUD, 2015, 2015 IEEE 8th International Conference on Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on Cloud Computing (CLOUD) 2015

the *intrusion* or even resources beyond the visibility of the virtualized environment that nevertheless share the same physical layer. Sensitive data may also be *inferred* by exploiting the virtualization introspection of a privileged process. For example, the VMM allows external observers to inspect VMs without interfering with them. Both intrusion and introspection problems are aggravated by virtualization-application scenarios that allow VM migration and virtualized-domain federation (e.g., cloud federations), where intruders who control a migrating VM might broaden their impact.

Deception: intentionally attempting to mislead other entities.

An unauthorized entity gains access to a system or performs a malicious act by posing as an authorized entity via: i) masquerading (i.e., spoofing, identity fraud), ii) falsifying data to deceive an authorized entity (e.g., wiretapping, reply attacks), iii) falsely denying responsibility for an act (i.e., repudiation), iv) misleading authorized users to perform actions that, alone or combined with the actions of other users, will have negative consequences on the system (e.g. social engineering attacks).

Deception is a common threat for physical systems, but in the case of virtualized environments, identity handling may be more difficult due, for instance, to the more complex and stratified hierarchical administration of privileges. As an example, at the virtual network level, when aggregating virtual networks into a federation, issues of role segregation and policy conflicts may arise, providing room for identity fraud. Moreover, the dynamics of adding and removing entities may be used by malicious entities to gain a new identity, for example, through inconsistencies in the migration process. Replay attacks are also facilitated by shared communication channels, which can be exploited at the virtual router level by replying to old control messages.[20] Concerning repudiation, the disposable nature of VMs, providing log features and the rollback procedures typical of virtualized environments, may have a strong impact on the non-repudiation of actions registered via logging.[21]

Disruption: causing failure or degradation of systems, negatively affecting the services they provide.

This threat may occur by i) directly *incapacitating* system components or communication channels, ii) *alternating or corrupting* system operations by adversely modifying system functions to induce delivery of corrupted data, iii) interrupting delivery of system services via *obstruction* by hindering system operations.

Incapacitation and alteration are typical disruptions for physical systems, where adversaries modify or corrupt crucial components to produce system degradation. Virtualized environments inherit these disruptions (e.g., uncontrolled allocation of resources), whereas the obstruction disruption is exacerbated due to the sharing of resources. For instance, physical resource overloading may cause degradation of a virtual network's performance, leading to disruption in communications, especially when the resources are located in the same area as the underlying network. We note that this may happen: i) unintentionally during the system's lifecycle (difficult to predict) or ii) maliciously in case of coordinated attacks.

Virtualized environments seek to cope with this severe class of threats by providing isolation solutions and by promoting fair distribution of resources among all virtualized entities (networking entities included).

[20] Fernandes NC, Duarte OCMB (2011) Xnetmon: A network monitor for securing virtual networks In: IEEE International Conference on Communications.. IEEE, Kyoto, Japan.

[21] van Cleeff A, Pieters W, Wieringa RJ (2009) Security implications of virtualization: A literature study In: International Conference on Computational Science and Engineering.. IEEE Computer Society, Washington, DC, USA.

However, these approaches are difficult to implement due to the intrinsic characteristics of virtualized systems that share computing resources and distribute them (possibly on demand) at runtime.

Usurpation: an unauthorized entity that may gain unauthorized control over a system.

Usurpation refers to *misappropriation* of identity via service, functionality, or data theft, or *misuse* of action (e.g., tampering), which causes a system component to perform a function or service that is detrimental to system security. For example, by sending a message that appears to have been originated from a privileged entity, an attacker may elevate its own privilege level to carry out a privilege escalation attack.

In virtualized environments, privilege escalation can be even more dangerous than in a physical environment because of multitenancy and the hierarchical structure of administrator privileges. In addition, VMM is a crucial target for usurpation-based misappropriation, due to its role in virtualization, as well as to the presence of vulnerabilities that allow guest-OS users the potential to execute arbitrary code on the host OS.[22]

2.2.2 Threat agents

As already stated in the glossary, a threat agent is "*someone or something with some capacity, a clear intention to manifest a threat and a record of past activities in this regard".* In a virtualized environment, it is crucial to be aware of which threats emerge or what ones might emerge from any particular threat-agent group. This study does not develop a new glossary on threat agents, but utilises the ENISA Threat Landscape 2013 consolidation of several publications.[23] The categorization of threat agent is as follows:

Corporations are organizations or enterprises that adopt or have been engaged in offensive tactics. In this context, corporations are considered hostile threat agents and their motivation is to gain competitive advantage over competitors, who also make up their main target. Depending on their size and industry, corporations usually possess significant capabilities, ranging from technology to human-engineering intelligence, especially in their area of expertise.

Cybercriminals are hostile by nature. Moreover, their motive is usually financial gain and, today, their skill level is typically quite high. Cybercriminals may be organized on a local, national or even international level.

Cyberterrorists have expanded their activities to engage in cyberattacks. Their motives may be political or religious, while their capability varies from low to high. Cyberterrorists' preferred targets are mainly critical infrastructure (e.g., public health, energy production, telecommunication), because their failures have severe impact on society and government. We note that, in the public material analyses, the profile of cyberterrorists still seems blurred.

Script kiddies are unskilled individuals using scripts or programs developed by others to attack computer systems and networks or deface websites.

---

[22] Common Vulnerabilities and Exposures (2012) CVE-2012-2450. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-2450.

[23] For example the Cyber Security Assessment Netherlands (see https://www.ncsc.nl/english/current-topics/news/cyber-security-assesment-netherlands.html), the Verizon report (see http://www.verizonenterprise.com/DBIR/).

Online social hackers (hacktivists) are politically and socially motivated individuals who use computer systems to protest or promote their cause. Their typical targets are high-profile websites, corporations, intelligence agencies, and military institutions.

Employees are among a company's staff, contractors, operational staff or security guards. They may have insider access to the company's resources and are considered both non-hostile threat agents (i.e., distracted employees) and hostile agents (i.e., disgruntled employees). Such threat agents possess significant knowledge that allows them to carry out effective attacks against their organization's assets.

Nation states have extremely high cyber-offensive capability and use it against adversaries. Nation states have recently become a prominent threat agent by deploying sophisticated attacks considered cyber weapons. The techniques and sophistication of these attacks show that nation states have a plethora of resources and extremely high skills, with expertise to access the most advanced techniques and personal know-how available. Such expertise often outstrips what is available from well-known public research institutes.[24]

All these agents may have an interest in exploiting certain vulnerabilities in virtual environments for different reasons. Only some specific threats typically originate with certain agents, such as the abuse of authorization involving corporation employees who use their administrator credentials to access systems.

2.3 Classification of weaknesses

The common weakness enumeration (CWE), maintained by MITRE[25], provides a common language for dealing with the causes of software-security weaknesses. CWE enables more effective discussion, description, selection, and application of the software-security tools and services used to counteract weaknesses in software systems. It also promotes better understanding and management of weaknesses related to software architecture and design. CWE defines a hierarchical structure that allows different levels of abstraction, from generic (i.e., *Injection* - CWE-74) to very specific (i.e., *OS Command Injections* - CWE-78) weaknesses. Some efforts have aimed at classifying weaknesses for a specific application domain. For example, in 2001, OWASP selected from CWE elements a set of weaknesses that affect web application security and classified them into the OWASP Top Ten. Since then they have released the ranked list on a periodically.[26] The National Vulnerability Database (NVD), a U.S. government repository of standards-based, vulnerability-management data, adopts a cross section of the CWE classification to categorize generic software vulnerabilities. The SANS Top 25 prioritizes generic weaknesses in the CWE by considering a software-system scenario.

Like the NVD, a cross section of the hierarchical CWE is considered for virtualization, which includes weaknesses having published vulnerabilities of interest, with special emphasis on those that affect any of the virtualization components identified in section 1. Weaknesses that affect generic ICT systems are also important and taken into account, since virtualization abstraction relies on standard software, firmware, and hardware components. In particular, weaknesses that target specific hardware features underlying a virtualization platform may play an important role, for instance CPUs with hardware-virtualization support

[24] Nation-state sponsored attacks happen rather often. For example according to Kaspersky Lab an infiltration in several of its internal systems and that the attack was believed to be sponsored by a Nation state. See http://www.crn.com/slide-shows/security/300077563/the-10-biggest-data-breaches-of-2015-so-far.htm/pgno/0/2

[25] https://cwe.mitre.org/

[26] OWASP Top Ten Project, https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

(e.g., Intel-VT) or physical security mechanisms (e.g., Trusted Computing and Trusted Platform Module), to name but a few.

Table 2.1 presents selected weakness groups in a table form. Each weakness group comes with a generic description (first row of each group in table 2.1) and virtualization specific remarks (second row of each group in table 2.1). We note that a complete group description is omitted here for brevity, but reported in annex A.

Table 2-1 of weakness groups. For each group, i) a general description and ii) virtualization-related peculiarities of the weaknesses considered are given.

WEAKNESS GROUP	DETAILS
Injection	This refers to weaknesses based on the lack of verification of user-controlled input. The injected data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
	In virtualized environments, injection issues still exist on interaction-specific virtualization components such as Hypervisor and VMM. They are often not well-tackled since the users involved frequently have administration-level permissions. Another specific type of injection is VM image injection or template injection.
Improper Authentication	This refers to weaknesses caused by incorrectly designed or implemented authentication mechanisms. It affects access-control policy specification and enforcement, including weaknesses involving authentication, authorization, user management, and communication between end-points.
	In virtualized environments, authentication applies both to end users and to system components. Examples of this such 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, to name but a few.
Management of credentials	This refers to weaknesses in how credentials are managed. It includes weaknesses in password management, such as lack of verification or enforcement of password strength, weak cryptography, password aging, weak password-recovery mechanisms for forgotten passwords, and the like. It also refers to insufficiently protected credentials, both at rest 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 transit. In addition, the lack of protection on credentials may leave room for advantages to be gained through personification. In virtualized environments, this may affect multiple levels of the virtualization stack.
Permissions and privileges management	This refers to weaknesses that involve managing permissions, privileges, and other security features used to enforce access control. Specifically, it includes issues caused by execution without the required privileges or incorrect privilege assignment, dropping or lowering errors, 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.
Cryptograp hic Issues	This refers to weaknesses related to the use of cryptography, in particular to cryptographic errors due to poor design or implementation. It includes plaintext storage or transmission of sensitive information, key-management errors like key exchange without entity authentication, and lacking or weak verification of expired keys. It also refers to weaknesses in cryptographic protocols and missing or weak encryption of sensitive data during storage or transmission (man-in-the-middle attacks).

	Virtualized environments exacerbate cryptographic issues by sharing of channels or resources. In particular, man-in-the-middle attacks become highly critical in virtualized environments, where messages from different tenants may share the same channel or infrastructure facilities.
Data handling	This refers to weaknesses in data-processing functionalities. A broad category, it includes string and type errors, generic representation errors like 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.
Information management errors	This refers to weaknesses that involve the improper handling of sensitive information. It specifically includes information exposure (a.k.a. 'information leak'), i.e., the intentional or unintentional disclosure of information to an unauthorized actor.
	In virtualized environments, attacks that exploit this weakness are more critical than in physical environments. For instance, side-channel attacks have recently been exploited in the virtualized environment,[27] which shows how an adversary might recover sensitive information by observing the behavior of a VM co-located on the same physical machine. 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.
Improper Input Validation	This refers to improper input validation, meaning that the system does not validate or incorrectly validates input. Specifically, it refers to pathname traversal and equivalence errors, 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, to name but a few.
	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.
Insufficient Verification of Data Authenticity	This refers to insufficiently verified data origin or data authenticity that leads to the acceptance of invalid data. It includes design and implementation weaknesses, such as the improper selection of data-authenticity mechanisms, the improper verification of cryptographic signatures, missing or improper validation of integrity checks and cross-site request forgery (CSRF). CSFR implies that the application does not, or cannot, sufficiently verify whether a well-formed, valid, and consistent request was provided by the user who submitted it.
	As for improper input validation, in virtualized environments, many components expose interfaces that may be targeted by attacks (e.g., CSRF) that exploit this weakness. Specifically, virtualization supports technology like the Intel-VT, secure crypto-processors, and Trust Computing (TC/TPM), which provide fundamental virtualization features but also open up a set of virtualization-technology-specific weaknesses (e.g., hypervisor blue-pilling rootkit in nested virtualization or misbehavior in authenticity-verification during boot). Finally, weaknesses in the cryptographic-issues group may also underlie insufficient verification of data origin and authenticity.

[27] Stelvio Cimato, Ernesto Damiani, Silvia Mella and Ching-Nung Yang, "Key recovery in public clouds: a survey on cross-VM side channel attacks," Second International Conference on Cloud Computing and Security (ICCCS 2016), Nanjing, China

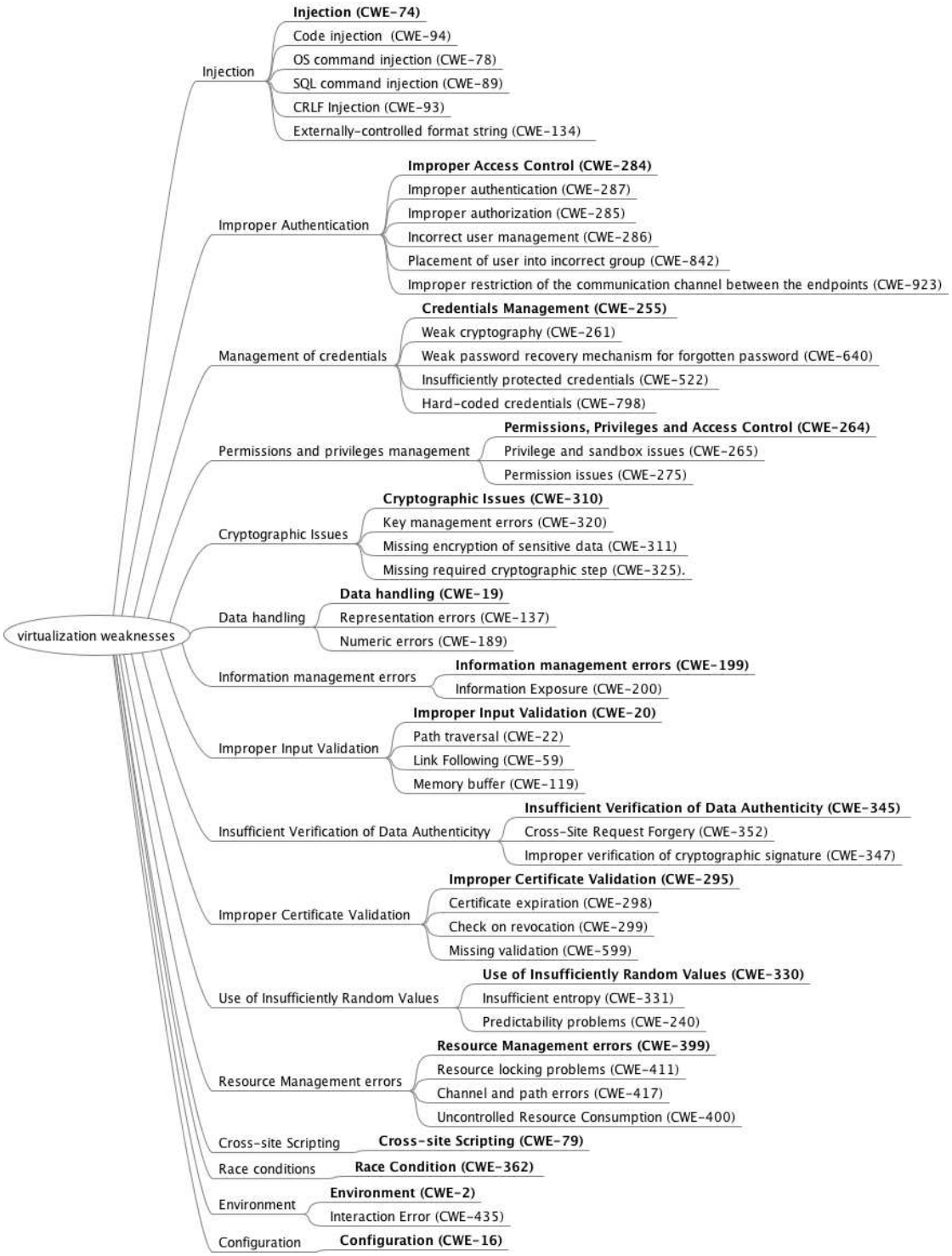
Improper Certificate Validation	This refers to a certificate that is not validated or incorrectly validated, possibly allowing man-in-the-middle attacks. It includes weaknesses related to improper validation with host mismatch, certificate expiration, revocation, or missing validation. It also includes weaknesses related to the improper verification of a 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 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.
Use of Insufficiently Random Values	This refers to weaknesses that involve generating predictable values in a context that requires unpredictability. This weakness is related to insufficient entropy in pseudo-random number generators (PRNGs), predictability problems, and the use of cryptographically weak PRNGs.
	It virtualized environments, this weakness is exacerbated by the virtualization of hardware devices. For example, achieving sufficient entropy is even more difficult since the virtualized environment reduces the quality of the source of entropy commonly adopted by PRNG algorithms.
Resource Management Errors	This refers to weaknesses that involve 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 weakness, such as the improper release of memory before removing the last reference, a.k.a. 'memory leak' or 'data-remanence issue'.
	In virtualized environments, this is crucial because many attacks are based on exhausting system resources to achieve denial of service 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.
Cross-site Scripting	This refers to user-controllable input that is not neutralized or is incorrectly neutralized before being placed in an output stream served to or used by other users. It mainly applies to traditional webpage surfing scenarios.
	In virtualized environments, there are dashboards used by clients to evaluate virtualization features or to inspect resources. These web-based dashboards allow interaction and thus must be protected against cross-site scripting.
Race conditions	This refers to code sequences that can run concurrently. Code sequences require temporary, exclusive access to shared resources. There are time windows in which the shared resources may be modified by code sequences that operate concurrently.
	In virtualized environments, the existence of numerous independently-managed, asynchronous components mandates carefully designing and implementing mechanisms to manage such situations.
Environment	This refers to weaknesses introduced during unexpected environmental conditions, mainly to technology-specific issues and interaction errors that occur when two entities *i)* work correctly when running independently and *ii)* interact in unexpected ways when running together.

	In virtualized environments, a number of 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 following weakness group, 'configuration.'
Configuration	This refers to weaknesses typically introduced during software-component configuration.
	Virtualized systems are often based on a number of 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.

Figure 2-1, below, maps the weakness groups in table 2-1 to CWE weaknesses, according to CWE hierarchy.

The following section first introduces vulnerabilities and their classification. We then use the components listed in section 1 as the starting point to investigate component-specific vulnerabilities that depend on the threats and the weakness groups they belong to.

Figure 2-1 Mapping between virtualization weakness groups and CWEs



2.4 Vulnerabilities in virtual environments

Common vulnerability exposure (CVE) is a standardized way to represent vulnerabilities, which enables us to unequivocally identify vulnerabilities and provides a unique ID with a standardized description. A vulnerability is a concrete implementation of a generic weakness in a specific software component or software ecosystem. CVE represents specific vulnerabilities in specific software packages, such as "Multiple cross-site scripting (XSS) vulnerabilities in Ansible UI before v2.0.5" (CVE ID: CVE-2015-1368), while CWE represents a generic weakness (i.e., a class of vulnerability) like "injection". CVE vulnerabilities point back to CWE weaknesses as a means to group vulnerabilities according to the weaknesses underlying them. Single CVE vulnerabilities may belong to more than one CWE, depending on their complexity. A vulnerability cannot necessarily be exploited in an attack. In some cases the vulnerability may provide some advantages for an attacker but cannot be exploited because of the contextual situation (e.g., a system is not accessible or its perimeter is protected). The CVE describes vulnerabilities by considering the effects and consequences of possible successful exploits.

Here, considering the threats and weaknesses in sections 2.2 and 2.3, and for each component described in section 1 (guest or host OS, containers as a special-case guest OS, hypervisor/VMM, virtual network, virtual storage), we present: i) the prominent vulnerabilities, ii) a table with a specific selection of vulnerabilities taken from CVE repositories, mentioning the threats and weaknesses they refer to, and iii) a set of consequences and, when available, examples of vulnerability exploits.[28] We note that some vulnerabilities are cross-component, exploiting a weakness on one component for the chance to exploit another weakness on a different component. For example, CVE-2012-1516 allows guest-OS users to cause denial of service (memory overwrite and process crash) or execute arbitrary code on the host OS, since the VMware hypervisor did not properly handle RPC commands. As a result, it can be listed as a weakness for both the guest OS and the hypervisor. When relevant, we will duplicate these cross-component vulnerabilities on the target components, explaining their corresponding peculiarities. Similarly, a single vulnerability may pertain to different threat categories. For example, the CVE-2010-0430 vulnerability, affecting RHEV-H, can cause a hypervisor to crash, leading to denial of service (disruption), but also allow local guests to obtain sensitive information from stack content (disclosure). Other vulnerabilities, such as privilege escalation, may lead to both usurpation and deception. When relevant, we will duplicate these multi-threat vulnerabilities by explaining the corresponding peculiarities. In the remainder of this section, we refer to CVE vulnerabilities using the corresponding CVE ID, which can easily be used to obtain the full description of the vulnerability from the official CVE repository.

2.4.1 Guest OS and Host OS

The guest and the host OSs share many weaknesses and vulnerabilities. For this reason, we have grouped them into the same section. Both guest and host OS are affected *by operating-system-level vulnerabilities*, which are considered very critical both because of the large number of vulnerabilities and because of their potential impact if administrator access is gained at the operating-system level. In addition, they share a number of *application-level vulnerabilities, as well as malware-based threats*. For improved protection, virtualization provides strategies to maintain isolation on guest and host OSs by avoiding the propagation of intrusions. However, the isolation can be bypassed by exploiting vulnerabilities that allow, for instance, evasion from the virtualized environment (e.g., VM escape). Communication between the guest OS and the host OS adopt a privileged channel even to perform some very basic activities like video display. Thus, any vulnerability at this level may give the guest the ability to execute code on the host, gaining information

[28] Public exploits are quite rare due to the security issues in revealing details on successful attacks.

from host memory,[29] or to gain privileges in the guest OS. Since both guest and host OSes are running instances of an image, any vulnerability of VM image handling, including the presence of inactive VMs is crucial. Virtualization environments foster handling of VM sprawl (i.e., excessive number of VMs), as well as snapshot management using image templates, tools, and repositories, whose vulnerabilities (e.g., VM image injection or information leakage) need to be considered. Furthermore, a VM image can also be migrated. This opens the door to vulnerabilities that target the image while in transit or while temporarily or permanently at rest in a shared repository. As with all virtualization components, the sharing of resources is one of the main sources of vulnerability, along with the use of monitoring or acting tools. At the VM level, vulnerabilities that lead to DoS often target the VM itself or the functionalities it provides. Some guest- or host-OS vulnerabilities can be used as an entry point to exploit resource-exhaustion weaknesses, for instance, via errors in input-validation weaknesses.

Vulnerabilities in the guest or host OSes involve nearly all the weakness groups like injection, privilege management, error in input validation, cross-site scripting, and those related to configurations and the environment. The following table shows a selection of vulnerabilities classified according to threat and weakness group.

Table 2-2 Threat / Weakness / Vulnerability table (Guest OS and Host OS)

THREAT	WEAKNESS	VULNERABILITY
Disclosure	Improper Input Validation Configuration	Side-channel attack[30] from one VM to another **Consequence:** the vulnerability could be used to reveal the existence of an application or a file on another virtual machine. See article "Software Side Channel Attack on Memory Deduplication"[31] with the description of attacks that take advantage of a difference in write access times on deduplicated memory pages in VM environment.
Deception	Improper Input Validation Configuration	Users gain privilege in guest OS **Consequence:** the vulnerability could be used by local users to upgrade their privileges in guest OS. CVE-2015-6933: VMware Tools package (HGFS, a.k.a. Shared Folders), installed in guest OS, allows Windows guest-OS users to gain privileges or cause denial of service though kernel-memory corruption.
Disruption	Injection	Injection to bypass security features (hypervisor protection) **Consequence:** the vulnerability could allow bypassing security features if an attacker runs a specially crafted application in guest OS. CVE-2016-0181: bypass of Hypervisor Code Integrity (HVCI) in Microsoft Windows 10 guest OS. The HVCI service determines whether code is executing in kernel mode, including drivers, is securely designed and trustworthy, securely allocates memory, and operates as intended.

[29] See http://www.blackhat.com/presentations/bh-usa-09/KORTCHINSKY/BHUSA09-Kortchinsky-Cloudburst-PAPER.pdf

[30] A side-channel attack is an attack based on information gained from the physical implementation rather than theoretical weaknesses

[31] See Kuniyasu Suzaki, Kengo Iijima, Toshiki Yagi, Cyrille Artho, Software Side Channel Attack on Memory Deduplication, National Institute of Advanced Industrial Science and Technology, RCIS, Tsukuba, Japan

THREAT	WEAKNESS	VULNERABILITY
	Improper Input Validation	Errors in input validation **Consequence:** the vulnerability could allow a local user to cause denial-of-service conditions. CVE-2015-1138: The manipulation in Apple OSX of an input to the hypervisor service leads to a denial-of-service vulnerability, affecting confidentiality, integrity, and availability.[32]
Usurpation	Privilege permissions Cross-site scripting Injection	Cross-site scripting (XSS) vulnerability **Consequence:** the vulnerability could allow remote attackers to inject arbitrary web script or HTML. Attackers elevate privileges and gain full control. They can then also access other customers' sensitive data. CVE-2015-1368: this vulnerability in a Red Hat component (Ansible virt-manager), which is present in many OS distributions,[33] allows remote attackers to inject arbitrary web script or HTML into various software modules, such as credentials, inventories, permissions, etc.[34]

The guest OS is the starting point for many attacks. Exploiting vulnerability in the guest or host OS may expose the system to the following consequences.

- *Attack the VM or other VMs (direct attack)*: an attacker can take advantage of relaxed access control or intentional inter-VM communications. This attack depends on host configuration and access control.
- *Attack the hypervisor*: this kind of attack usually starts in a guest OS and is hypervisor-dependent. Paravirtualized drivers, clipboard sharing, display output, and network traffic tend to create this type of channel.
- *Attack the hardware on the host*: hardware platforms often request firmware updates. By accessing this mechanism from a VM, an attacker could upload rogue firmware to favor the attack. For this reason, many hypervisors filter such commands if possible.[35]
- *Attack the host architecture*: this is the typical side-channel attack against a shared component. Examples of this kind of exploit include use-after-free or double-allocation attacks, as well as the rogue use of a memory-ballooning system.[36]

The above consequences were reported in a few public available exploits of vulnerabilities. For instance, an exploit of CVE-2015-1368 (Privilege Escalation & XSS & Missing Authentication[37]) was achieved by targeting

---

32 See Apple https://vuldb.com/?id.74720

33 The component virt-manager is present in the following distributions: Arch Linux, Debian, Fedora, Frugalware, Gentoo, Mandriva Linux, NetBSD, OpenBSD, openSUSE, PC-BSD, Red Hat Enterprise Linux, Ubuntu, Scientific Linux

34 See Packet Security https://packetstormsecurity.com/files/129944/Ansible-Tower-2.0.2-XSS-Privilege-Escalation-Authentication-Missing.html

35 For example VMware ESXi filters microcode updates when they originate from a VM.

36 The "use-after-free" vulnerability means referencing memory after it has been freed. The memory ballooning system attack makes a ballooning system believe some memory can be shared when in fact it should be private.

37 See https://www.exploit-db.com/exploits/35786/

the dashboard of Red Hat Ansible[38] installed in a VM. It allowed the attacker to elevate its privileges and gain full control over the VM by accessing sensitive data. Another recent exploit, called Flip Feng Shui,[39] targets the memory pages on a victim VM that runs on the same host as the attacker VM, relying on a refinement of the hardware vulnerability known as the Rowhammer bug. The attack can be summarized in three phases: i) the attacker VM profiles its memory to find memory cells vulnerable to the Rowhammer bug, ii) the attacker writes a memory page known to exist in the victim on the vulnerable memory location, iii) the attacker triggers Rowhammer to modify the victim's memory.

2.4.2 Containers

In a container-based scenario, an application's operating environment is virtualized. The result is an isolated container in which the application can run.[40] Examples of this scenario include application virtualization, Linux containers, FreeBSD-style jails, and sandboxing. Containers are different from guest OS and virtual machines in general, but they share some security vulnerabilities, such as access control, privilege permissions, and coding weaknesses. Specifically, container vulnerabilities mainly aim to escape from container isolation, similar to VM escape, or to escalate privileges in the container software layer (e.g., symbolic link traversal on container respawn). Privilege escalation in containers is considered very risky, since it allows gaining the same privilege level on the host OS (e.g., container breakouts) mainly because not all resources at container level are namespaced (a namespace uniquely identifies a set of names so that there is no ambiguity when objects having different origins but the same names are mixed together). For instance, in a kernel keyring for handling cryptographic keys, keys are generally separated by a UID not normally namespaced at the container level, so users with the same UID may gain access to the keys. Other vulnerabilities related to weak permissions or improper authorization permit local users to obtain sensitive information, to perform protocol downgrade attacks via a crafted image or to edit their profiles. As with a VM, image management and protection is crucial i) for preventing attackers from running maliciously poisoned images, putting that host and data at risk, and ii) for keeping the image updated to avoid known vulnerability exploits. Proper management of credentials is critical in all virtualization components (i.e., compromised secrets). It is even more vital with containerization applied to micro-service architectures, where containers are continuously started and stopped. Specifically related to containerization, every kernel-related vulnerability (called kernel exploit) is magnified by the fact that, unlike in case of a VM, the kernel is shared among all containers and the host. Thus, any kernel issue raised at the container level has great potential impact, even leading to DoS when exhausting certain specific kernel-level resources.[41]

In the following table, we present a selection of vulnerabilities classified according to threat and weakness group.

38 See http://www.ansible.com/tower

39 Razavi, Kaveh, et al. "Flip Feng Shui: Hammering a Needle in the Software Stack." (2016).

40 See Michael Pearce (University of Canterbury) et al. "Virtualization: Issues, Security Threats, and Solutions", 2013

41 https://gallery.mailchimp.com/979c70339150d05eec1531104/files/Docker_Security_Red_Hat.pdf

Table 2-3 Threat / Weakness / Vulnerability table (Containers)

THREAT	WEAKNESS	VULNERABILITY
Disclosure	Improper Access Control Management of Credentials	Weak permissions **Consequence:** this vulnerability allows local users to modify the host, obtain sensitive information or perform protocol downgrade attacks via a crafted image. CVE-2015-3630: Docker engine uses weak permissions in software modules**.**
Deception	Improper Access Control	Improper authorization and change of profile **Consequence:** this vulnerability allows users to modify their profiles. CVE-2014-6408: Docker allows remote attackers to modify the default run profile and bypass the container by applying improper security options.
Disruption	Permission and Privilege Management	Containerization escape **Consequence:** this vulnerability allows local users to escape containerization. CVE-2015-3629: A library used in Docker engine allows local users to escape containerization ("mount namespace breakout") and cause disruption. It can also lead to write-to-arbitrary-file on the host system.
Usurpation	Permission and Privilege Management	Privilege or profile escalation in container software layer **Consequence:** this vulnerability allows local users to gain privileges. CVE-2016-3697: C libraries in Docker improperly treat a numeric UID as a potential username in the password file in a container. CVE-2015-3629: Container respawn (recreation) allows local privilege escalation. CVE-2015-3627: Insecure opening of file-descriptor leads to privilege escalation.

Exploiting vulnerability in the container can expose to the following consequences.

- *Attack the application*: this mainly leads to disclosure of data.
- *Attack the whole container*: this can lead to denial of service and disruption of the services provided.
- *Attack the host architecture*: this could lead to arbitrary file modification on the host system and execution of code (disruption).

The above consequences have been reported in some well-known vulnerability exploits. For instance, an exploit based on a vulnerability in sysfs call in the container that does not correctly support namespaces allows root users inside the container to execute commands at the host-OS level while maintaining root privileges.[42] Another exploit, called shocker,[43] demonstrates an information-leakage weakness in Docker

42 Before 1.0: Breakout by design via sysfs – http://blog.bofh.it/debian/id_413

43 https://medium.com/@fun_cuddles/docker-breakout-exploit-analysis-a274fff0e6b3#.5rgxeia4z

containers. It is based on direct inode indexing, instead of protected pathname, to give unauthorized access to privileged filesystem data. As an additional example, a recent exploit of a container vulnerability allowed the uncontrolled download of the full source code of Vine, a Twitter service, plus, its API keys as well as the third-party keys.[44]

2.4.3 Hypervisor / VMM / Management server and console

The hypervisor, the VMM and their management tools share weaknesses and vulnerabilities. For this reason, we grouped them into the same section. When not specifically mentioned, we refer to both hypervisor, VMM, and the corresponding management server and console as hypervisor. From a security perspective, the ideal hypervisor is a system with the fewest possible lines of code (i.e., reduced attack surface). In other words, a minimal hypervisor delivers all its basic functionalities (managing hardware for guest VMs, maintaining a secure environment and isolation between the VMs) with no need for extra services such as plug-n-play, wireless services or graphical user interface, increasing its security and reducing the number of exploitable vulnerabilities. However, the hypervisor is a complex and fundamental component in any virtualized system and is thus the most attractive target for attack. As a result, it shows a large number of vulnerabilities, mainly related to its fundamental security features: i) VM isolation, which is the target of the majority of attackers, ii) internal software-based channels for communication with VM (e.g. VM exits), which are suitable intrusion channels for any attacker, and iii) VMM or additional API for inspection, which are vulnerable like any software-system entry point (e.g., XSS on the web interface is available).

Many hypervisor vulnerabilities leverage weaknesses in other components, mainly at the host- or guest-OS level, to gain enough privileges to directly target the hypervisor itself. Some vulnerabilities focus on reading data from the hypervisor or other guests by exploiting classical software weaknesses at the hypervisor-call (a.k.a. hypercall) level, such as data-handling issues, authentication, and privilege issues. The effects of these vulnerabilities include DoS that could corrupt hypervisor memory space and possibly execute arbitrary code, obtain sensitive information from hypervisor stack content or gain privileged access to the hypervisor. Some vulnerabilities are context specific and require a specific contextual situation to become exploitable. Environmental context is fundamental at the hypervisor level. For instance, when physical CPUs are saturated, bugs on specific hypercalls may lead to buffer-overflow issues (CVE-2014-1895). In general, hypercall vulnerabilities require an attacker to control the guest OS and exploit hypercall-specific vulnerabilities such as passing malicious parameters (CVE-2009- 3290).

Hypervisors are also vulnerable to a set of injection-related vulnerabilities like VMM-oriented injection, OS-command injection or software-library injection, especially when a special mode, like shadow-mode paging or nested virtualization, is activated (CVE-2016-1571). Hyperjacking focuses on injecting a hypervisor from an OS into the hardware that converts a physical system to a virtual machine. It is related to vulnerabilities at the driver level that allow reading raw disks and at the boot level that allow boot-files manipulation. Compared to physical systems, the presence of VM images, snapshots, and a VM hard disk facilitate the hyperjacking.

Hypervisors also show vulnerabilities when providing specific functionality for VM management, such as VM migration, including cloning. VM-cloning may expose the virtualization system to a number of vulnerabilities, since it is obtained very easily via copying and pasting VM-image files. Even more dangerous are issues related to VM templates used by hypervisors as basic images for VM provisioning. The hypervisor writes hypervisor mappings to certain shadow-page tables when live migration is carried out, allowing local guests to read or write to invalid memory, thus causing DoS (CVE-2013-4356). The VM-escape vulnerabilities treated in section 2.4.3 are normally considered part of the set of vulnerabilities that

[44] See Hacker News website http://thehackernews.com/2016/07/vine-source-code.html

affect hypervisors, since some of them rely on weaknesses at the hypervisor level. In the case of the path-traversal vulnerability, the abuse of VM guest tools allows access to the host machine from the guest machine. For instance, VM Drag-n-sploit abuses of the drag-and-drop communication channel granted by the hypervisor might inject commands.[45]

Hypervisors are also vulnerable to some attacks on the virtualization of hardware including device virtualization, virtual CPUs, memory management, and interrupt handling, to name but a few.[46] Some of these vulnerabilities are based on firmware rootkits that can open a back door for an attacking VM to access all other VMs or, in general, open physical driver-related vulnerabilities. Hypervisors handle virtual CPUs assigned to different hosts. Due to this central role, weaknesses in virtual CPU handling makes way for a number of vulnerabilities that allow reading of virtual CPU registers and the like. In general, any activity that manages different VMs' memory, including the memory used by hypervisors, is subject to vulnerabilities. As an example, MMU maintains shadow-page tables for each guest VM. When not handled properly with virtualization-aware hardware, MMU may allow information leakage (CVE-2010-0298). While emulating interrupts, the hypervisor uses some specific data structures that may be a target of attacks, making the VMs unstable or unusable. Some specific communications from guest OS to hypervisor are strictly related to the hardware platform, as well. This is the case when the hypervisor intercepts operations invoked by a guest that require its intervention with root privileges. The vulnerabilities related to these issues are highly platform-dependent.

From the resource-management point of view, hypervisors are responsible for fairly sharing resources. A number of vulnerabilities focus on resource hugging and resource contention with the aim of putting the virtualized system in stressed conditions, thus easily exploiting additional weaknesses or generating DoS.

In the following table, we present a selection of vulnerabilities classified according to threat and weakness group.

Table 2-4 Threat / Weakness / Vulnerability table (Hypervisor / VMM)

THREAT	WEAKNESS	VULNERABILITY
Disclosure	Improper Input Validation	Input validation in hypervisor software **Consequence**: this vulnerability allows local hardware-virtual-machine (HVM) guests to read data from the hypervisor or other guests. It can also cause a denial of service (host crash). CVE-2014-7188: in Xen hypervisor the intercept function in a software library uses an improper range.
	Data handling	Data handling issue due to off-by-one error (i.e. an iterative loop iterates one time too many or too few) in software function **Consequence**: this vulnerability allows local users to obtain sensitive information from hypervisor memory (or cause a denial of service/host crash).

[45] Shackleford, Dave. Virtualization security: protecting virtualized environments. John Wiley & Sons, 2012.

[46] Perez-Botero, Diego, Jakub Szefer, and Ruby B. Lee. "Characterizing hypervisor vulnerabilities in cloud computing servers." Proceedings of the 2013 international workshop on Security in cloud computing. ACM, 2013.

THREAT	WEAKNESS	VULNERABILITY
		CVE-2014-1895: this vulnerability affects Xen hypervisor.
		Data handling in memory (data-handling issue due to stale data[47] in a segment register) **Consequence**: this vulnerability allows local guests to obtain sensitive information from the hypervisor stack content. For example, guest-OS users can read from or write to arbitrary memory by modifying the address that is used for memory mappings. CVE-2010-0430: vulnerability in Xen hypervisor. CVE-2013-4368: another similar vulnerability in Xen hypervisor.
Deception	Improper Certificate Validation Permission and Privilege Management	Bypassing security restrictions **Consequence:** the platform is prone to bypasses of intended protection and to privilege escalation. CVE- 2014-0092: a component of the Red Hat Enterprise Linux / WebSphere Application Server Hypervisor suite could allow a remote attacker to bypass security restrictions, caused by an error when verifying unspecified certificates. By persuading a victim to visit a specially-crafted website, an attacker could exploit this vulnerability to bypass certificate-validation checks and gain access to the system.[48] CVE-2013-2196: example of VM template cloning issue on Xen Server PV (Pure Virtual) guests that "*allow local guest administrators with certain permissions to have an unspecified impact via a crafted kernel which result in privilege escalation*"
	Race conditions	Race conditions exploit a small window of time between the application of a security control and the execution of the service. **Consequence:** guest users gain privileges or cause a denial of service. CVE-2010-0419: This refers to a bug that permits malicious Ring-3 processes to execute privileged instructions when SMP is enabled, because of the presence of a race-condition scenario.
	Improper Input Validation Configuration Improper Access Control	Wrong implementation of memory-management in virtualization technologies **Consequence:** the vulnerability could cause application bugs that are not exploitable when running the application in non-virtualized operating systems to become exploitable when running the application within a guest OS. This vulnerability can cause deception, as well as disclosure, disruption, and usurpation. CVE-2010-1225: Windows Virtual PC does not properly restrict access from the guest OS. Windows 7 relies on Virtual PC technology to implement the backward-compatibility XP Mode for legacy-Windows

[47] In computer processing, if a processor changes the value of an operand and then, at a subsequent time, fetches the operand and obtains the old rather than the new value of the operand, then it is said to have seen stale data.

[48] See https://www.ibm.com/blogs/psirt/ibm-security-bulletin-ibm-websphere-application-server-hypervisor-gnutls-certificate-security-bypass-cve-2014-0092/

THREAT	WEAKNESS	VULNERABILITY
		applications. An application running on Windows 7 in XP Mode may be exploitable while the same application running directly on a Windows XP SP3 system is not.[49]
	Cross-site scripting	Cross-site scripting attack on an administration console **Consequence:** stealing a victim's authentication cookies. CVE-2008-3253: describes a cross-site scripting attack on a remote administration console that exposed all of Xen's VM management actions to a remote attacker.
Disruption	Improper Input Validation Resource Management errors	VM escape: due to hypervisor coding or management errors, an attacker runs code on a VM and interacts directly with the hypervisor. **Consequence:** the vulnerability allows local guest users to cause denial of service (out-of-bounds write and guest crash) or execute arbitrary code. CVE-2015-3456: This is the so-called VENOM bug (Virtualized Environment Neglected Operations Manipulation).[50] This vulnerability is in the floppydisk-controller (FDC) code used in many systems (Xen, KVM, VMware, Microsoft, etc.). It is also agnostic about guest operating system. An attacker (or an attacker's malware) would need to have administrator or root privileges in the guest operating system in order to exploit it[51] CVE-2012-1516: The VMX process in VMware ESXi does not properly handle RPC commands, which allows guest-OS users to cause denial of service (memory overwrite and process crash) or execute arbitrary code on the host OS.
	Injection	Injection in hypervisor software libraries (when a special mode, such as shadow-mode paging or nested virtualization, is enabled) **Consequence**: this vulnerability allows local guest users to cause denial of service (host crash) via a non-canonical guest address. CVE-2016-1571: This vulnerability affects Xen and hardware virtual machines running in Xen (HVM is how Xen calls the hardware-assisted virtualization technology). It allows local HVM guest users to cause a denial of service (host crash) via a non-canonical guest address in a certain instruction, which triggers a hypervisor bug check.
	Injection Improper Input Validation	OS command Injection **Consequence**: host-OS users gain privileges on the guest OSes. CVE-2010-4297: OS-command-injection issue due to input validation of VMware Tools update in Vmware ESX/ESXi.

49 See Core Security https://www.coresecurity.com/content/virtual-pc-2007-hypervisor-memory-protection-bug

50 See Crowd Strike http://venom.crowdstrike.com

51 See Intel https://kc.mcafee.com/corporate/index?page=content&id=SB10118

THREAT	WEAKNESS	VULNERABILITY
	Data Handling Resource-management Errors	Memory control failure in hypervisor code **Consequence**: this vulnerability allows attackers to execute arbitrary code and to cause a hypervisor crash. CVE-2014-3124: this vulnerability affects Xen systems[52] where a malicious administrator of a domain privileged with regard to a guest can cause Xen to crash, leading to denial of service.
		Data handling when dereferencing a bogus address **Consequence**: this vulnerability allows guest users to cause denial of service. CVE-2011-2519: this vulnerability affects Xen hypervisor.
		Issue with hypervisor firmware **Consequence**: this vulnerability allows local users to affect system availability. CVE-2013-3838: this vulnerability affects Oracle SPARC hypervisor firmware.
		Hypervisor mappings to certain shadow-page tables when live migration is performed **Consequence**: this vulnerability causes denial of service (it also allows local guests to read or write to invalid memory). CVE-2013-4356: this vulnerability affects Xen hypervisor.
	Injection	Injection issue (unexpected values) **Consequence**: this vulnerability allows attackers with access to a guest operating system to crash the host operating system, effectively denying services to legitimate users. CVE-2014-8866: injection issue due to unexpected values allowed for registers holding hypercall arguments controlled by guest software; guests can cause denial of service (host crash) in a Xen hypervisor.
	Configuration	Use of Python exec() statements to process the custom kernel's user-defined configuration file **Consequence**: triggers the destruction of another co-hosted domain. CVE-2007-4993: Xen's bootloader for paravirtualized images left open the possibility of executing arbitrary python code inside Dom0.
Usurpation		Privilege escalation in the guest OS

[52] See Xen documentation http://xenbits.xen.org/xsa/advisory-92.html

THREAT	WEAKNESS	VULNERABILITY
	Permission and Privilege management	**Consequence:** this vulnerability allows local users to gain privileges in the VM (the vulnerability also allows unauthorized disclosure of information and allows unauthorized modification and disruption of services). CVE-2015-7078: "use-after-free" vulnerability (referencing memory after it has been freed) in hypervisor component when using Apple OS X, which allows local users to gain privileges in operations involving VM objects. CVE-2015-6933: kernel-memory-corruption vulnerability. Successful exploitation of this issue could lead to an escalation of privilege in the guest operating system.
		Privilege escalation in host OS **Consequence:** the vulnerability allows host-OS users to gain host-OS privileges CVE-2016-2077: vulnerability in desktop-virtualization software.

Most of the attacks on the hypervisor or on the VMM originate from hypervisor/VMM coding errors but are usually initiated in the guest OS. Exploiting vulnerabilities in the hypervisor/VMM/management server and console can expose systems to the following consequences.

- *Attack on the VMs hosted by the hypervisor*: this can lead either to disclosure of information from the VMs or full/partial disruption of system availability. The weaknesses exploited in this class of threats are validations of inputs and certificates, privilege escalation, and data-handling issues.
- *Attack on the host platform*: this mainly leads to denial of service (disruption). This kind of problem is sometimes due to issues related to hardware and firmware. The VENOM (Virtualized Environment Neglected Operations Manipulation) bug[53] is a "guest-escape" bug in popular opensource code that spawned a number of commercialized virtualization products, such as KVM, Xen and VirtualBox. It allows a buffer overflow in the software component that simulates floppydisk drives at the hypervisor level. Attackers inside any guest VM could get data and code from their siblings by digging into the host-operating-system memory space.

The above consequences were experimented in some concrete exploits of vulnerabilities, such as Virtual PC Hypervisor Memory Protection[54] and Cloudburst VM escape.[55]

In the Virtual PC Hypervisor Memory Protection exploit, the target was the backward-compatibility XP Mode for legacy-Windows applications provided by Windows 7, specifically the CVE-2010-1225 vulnerability in VMM memory management, a bug that makes standard Windows anti-exploitation mechanisms ineffective.

53 See https://nakedsecurity.sophos.com/2015/05/14/the-venom-virtual-machine-escape-bug-what-you-need-to-know/

54 See https://www.exploit-db.com/exploits/11786/

55 See http://searchcloudsecurity.techtarget.com/definition/Cloudburst-VM-escape

Cloudburst, is an exploit allowing VM escape and enabling a guest VM to attack its host.[56] The method takes advantage of a flaw in VMware Workstation, when working in conjunction with Cloudburst, IBM's cloud-service-provisioning software for cloud providers.

2.4.4 Virtual networks

Virtual networks connect VMs the same way as physical networks. They are affected by vulnerabilities similar to those that target traditional physical networks, such as personification, ARP poisoning, and congestion, to name but a few. They also share threats with SDN.[57] Virtual-networking-specific vulnerabilities focus primarily on networking emulation or on any virtual-networking device.

Network management is emulated via software by the hypervisor. Virtualized networks thus suffer from a number of software-related vulnerabilities aimed at modifying their networking functionalities, enabling information leakage or corruption. In addition, because virtualized networks share the underlying physical network, uncontrolled allocation of resources may generate an unpredictable overload. Similarly, latency attacks[58] can be implemented in software-defined networks, where authorized configurations of components (e.g., routing-table configuration, bandwidth modification) and environment re-configuration (e.g., VMs migration) can be implemented to deliberately increase network latency. This attack can either cause a denial of service or penalize target users vis-à-vis competitors. Considering virtualized-networking devices, there are several vulnerabilities that generate misbehaving virtual routers, for instance, enabling virtual routers to send old control messages multiple times, thus implementing replay attacks.[59] Since virtual routers are based on runnable images to be executed by the hypervisor, any injection or modification of virtual router images can be obtained by exploiting the vulnerabilities in image management at the hypervisor level. In addition, since virtualization abstraction relies on physical networking facilities, the virtual device and its communication with the real driver may be exposed to vulnerabilities such as the hot-unplug QEMU bug (CVE-2011-1751) used to develop the Virtunoid attack.[60] The hypervisor assigns multi-homed virtual network-interface card on a VM to improve connectivity. This features opens up a vulnerability known as NIC escape, whereby an attacker implements a bridge between the virtual interfaces and gains desired network access.[61] In the following table, we present a selection of vulnerabilities classified according to threat and weakness group.

---

56 See http://searchcloudsecurity.techtarget.com/definition/Cloudburst-VM-escape

57 See ENISA https://www.enisa.europa.eu/publications/sdn-threat-landscape

58 C.A. Ardagna, E. Damiani, "Network and Storage Latency Attacks to Online Trading Protocols in the Cloud," in Proc. of the International Conference on Cloud Computing, Trusted Computing and Secure Virtual Infrastructures, Amantea, Italy, October, 2014

59 Bays, Leonardo Richter, et al. "Virtual network security: threats, countermeasures, and challenges." Journal of Internet Services and Applications 6.1 (2015): 1.

60 N. Elhage. Virtunoid: Breaking out of KVM. nelhage.com/talks/kvm-defcon-2011.pdf, August 2011.

61 Michael Pearce, Sherali Zeadally, and Ray Hunt, "Virtualization: Issues, security threats, and solutions," 2013.

Table 2-5 Threat / Weakness / Vulnerability table (Virtual networks)

THREAT	WEAKNESS	VULNERABILITY
Disclosure	Information-management Errors	Uncontrolled handling of sequential requests for virtual networks.[62] **Consequence**: allows reconstructing physical topology of the underlying network. Preliminary to network topology poisoning attacks.[63]
	Information-management Errors Improper Access Control	Inspecting virtual network traffic or accessing virtual router.[64] **Consequence**: Obtaining confidential routing information from virtual network and possibly to physical network. CVE-2013-6398: the virtual router allows remote attackers to bypass intended restrictions.
Deception	Injection	Injection of malicious messages making other entities in the virtual network believe such messages come from another entity. Federation, dynamic adding and removing of nodes, and migrations exacerbate this vulnerability in virtual environments.[65] **Consequence:** Identity fraud.
	Information management errors Data Handling	Rollback of networking activity log stored in a VM. **Consequence:** loss of network entity activities with possible impact on non-repudiation of actions. CVE-2012-3449**:** writable permissions in open switch implementation allow local users to delete and overwrite arbitrary files.
Disruption	Resource-management Error Improper Access Control	Software-controlled latency over virtualized networks. **Consequence:** network-level denial of services
	Insufficient Verification of Data Authenticity	Misbehaving virtual routers send old control messages multiple times (reply attacks). **Consequence:** corruption of data plane and denial of service.
	Resource-management Error	Uncontrolled allocation of resources of multiple virtual networks on the same substrate as physical network. Coordinated attack or unintentional overload.

62 Pignolet Y-A, Schmid S, Tredan G. , Adversarial vnet embeddings: A threat for isps?. In: IEEE INFOCOM. IEEE, Turin, Italy 2013.

63 Hong, Sungmin, et al. "Poisoning Network Visibility in Software-Defined Networks: New Attacks and Countermeasures." NDSS. 2015.

64 Fukushima M, Sugiyama K, Hasegawa T, Hasegawa T, Nakao A. Minimum disclosure routing for network virtualization and its experimental evaluation. IEEE/ACM Trans Netw PP(99):1839–1851, 2013

65 van Cleeff A, Pieters W, Wieringa RJ., Security implications of virtualization: A literature study. In: International Conference on Computational Science and Engineering. IEEE Computer Society, Washington, DC, USA, 2009

THREAT	WEAKNESS	VULNERABILITY
		Consequence: degradation of performance
	Improper Validation	Wrongly throwing exceptions when handling malformed, truncated, or maliciously-crafted packets, causing switch disconnection because an exception occurred in an I/O thread. **Consequence:** denial of service (DoS) CVE-2015-1166: packet deserialization. CVE-2015-7516: exceptions in application-packet processors. CVE-2016-2074*:* buffer overflow in Open vSwitch software library.
Usurpation	Injection	Injection of malicious messages from a fake source with high privileges. **Consequence**: privilege escalation.
	Privileges and Permissions	Improper handling of identities and associated privileges **Consequence:** this vulnerability allows controlling virtual network nodes like virtual routers.
	Credentials Management	Network-management console access using brute-force password-guessing **Consequence:** empowers attacker to launch attacks on network**.**

Exploiting virtual network vulnerability can expose systems to the following consequences.

- *Network intrusion*: sensitive data can be grabbed by circumventing system security protections.
- *Data interception* is a common threat in networking environments but its effects are further exacerbated in virtual environments where the physical level is shared (e.g., topology inference).

In virtual environments, these kinds of threats are not easy to confine, because intruders may obtain privileges on other resources in the same physical layer as well, even if they are not the direct target of the intrusion. In addition, in virtual networks, disruption-related threats are especially critical because i) they affect both the physical layer and virtualized resources ii) they may be side-effects of attacks on other assets that share the same physical layers or directly on a crucial physical node, and iii) recovery and relocation of resources can cause resource overload. Finally, being programmable, virtual network behaviour can be transparently modified by administrators through network-component reconfiguration, leading to deception and disruption threats (e.g., latency-based attacks). Virtualization environments (including the cloud) make it easier to get away with some of the above attacks, since they reduce upfront attack costs or make it possible to hide the attacker's tracks from auditing and forensics analysis.[66]

There are documented attacks to gain access to Openflow-based network controllers[67] in different steps: i) how to discover the virtual network, ii) how to perform reconnaissance by identifying targets (e.g., ACLs,

[66] C.A. Ardagna, E. Damiani, "Network and Storage Latency Attacks to Online Trading Protocols in the Cloud," in Proc. of the International Conference on Cloud Computing, Trusted Computing and Secure Virtual Infrastructures, Amantea, Italy, October, 2014

[67] See Abusing Software Defined Networks in Black Hat website https://www.blackhat.com/docs/eu-14/materials/eu-14-Pickett-Abusing-Software-Defined-Networks-wp.pdf

sensors) using various toolkits, iii) how to obtain credentials through an exploit and gain access to the controller.

2.4.5 Virtual storage

In networked environments, storage virtualization has brought multiple benefits, and virtual machines and applications are not bound to specific physical hardware (e.g., traditional network attached storage – NAS – and storage area network – SAN). However, this new virtualized-storage architecture suffers from common threats, such as disclosure and disruption. From a technological perspective, many issues arise from merging different vendors' software components[68] onto a single platform. Major issues in storage virtualization involve access control, authentication, and credential management. Vulnerabilities there focus on bypassing access control or gaining permissions at the file-system level. Virtual storage can be also the target of DoS attacks that aim to artificially increment the physical hard drive consumption.

Additional vulnerabilities are generated by the presence of web-based interfaces or storage management GUIs. Yet other vulnerabilities stem from facilities provided by virtual storage handlers like backup approaches and replicas.

Table 2-6 Threat / Weakness / Vulnerability table (Virtual storage)

THREAT	WEAKNESS	VULNERABILITY
Disclosure	Cross-site Scripting Improper Input Validation Injection	Cross-site scripting (XSS) vulnerability in Storage Manager for Virtual Environments, caused by improper validation of user-supplied input. **Consequence:** this vulnerability allows remote authenticated users to inject arbitrary web scripts or HTML via a crafted URL. For example, a remote attacker could exploit this vulnerability using a specially-crafted URL to execute a script in a victim's web browser within the security context of the hosting web site, once the URL is clicked. The attacker could use this vulnerability to steal the victim's cookie-based authentication credentials. CVE-2015-1988*:* injection of arbitrary web script or HTML via a crafted URL.[69]
	Management of Credentials Improper Access Control	Access and privilege-escalation vulnerability in the Storage Manager GUI for Virtual Environments **Consequence:** this vulnerability allows remotely authenticated users to obtain sensitive information by reading the VM inventory and a single VM's storage. CVE-2015-7429: select an existing virtual machine from the VM inventory, perform a restore operation (without overwriting the existing virtual machine, if already running) and access unencrypted storage inventory.[70]

[68] For example the IBM Tivoli Storage Manager for Virtual Environments integrates the VMware vSphere GUI: for this integration, the NIST NVD lists several vulnerabilities.

[69] See IBM report http://www-01.ibm.com/support/docview.wss?uid=swg21967532

[70] See IBM report http://www-01.ibm.com/support/docview.wss?uid=swg21973087

THREAT	WEAKNESS	VULNERABILITY
		Authentication vulnerability in Virtual Storage Appliances (iSCSI Backup Systems) **Consequence:** this vulnerability allows remote unauthorized access to the storage appliance. CVE-2013-6211: remote unauthorized access is allowed.[71]
	Management of Credentials	Cryptographic issues and password-encryption weakness **Consequence:** this vulnerability makes it easier for context-dependent attackers to obtain clear-text passwords via a brute-force attack. CVE-2014-4623: use of DES-encryption for password hashing.[72]
Deception	Insufficient Verification of Data Authenticity	Authentication issue such as an attack on SSO in a (virtualized) environment **Consequence:** this kind of vulnerability could originate fraud or identity theft. CVE-2016-3686: a flaw lets remote users obtain session ID, since this information is included in some URL query parameters.[73] CVE-2015-6853 and CVE-2015-6854: due to insufficient verification of requests, this vulnerability may allow a remote attacker to gain access to the system.[74]
Disruption	Management of Credentials Improper Access Control	Authentication vulnerability in Storage Manager for Virtual Environments **Consequence:** local users can cause a denial of service or disk consumption via GUI actions. They could also obtain sensitive VM data. CVE-2013-6713: authorization for backup and restore operations is not properly checked, enabling users to perform the following actions, regardless of their specific level of authorization: accessing all data within the VMs that have been backed up previously or backing up and accessing data that has not previously been backed up or spawning multiple restores, which could exhaust storage.[75]
		Unauthorized access vulnerability in Storage Manager GUI for Virtual Environments **Consequence:** this vulnerability allows remote attackers to obtain administrative privileges to disrupt services. It can also cause access to confidential information. CVE-2015-7425*:* crafted-URL pattern that triggers back-end function execution. An attacker could adversely affect system operation,

71 See HP Bulletin https://packetstormsecurity.com/files/125921/HP-Security-Bulletin-HPSBST02968-2.html

72 See Security Focus report http://www.securityfocus.com/bid/70732/solution

73 See Security Tracker http://www.securitytracker.com/id/1035519

74 See CA Technologies Support http://www.ca.com/us/support/ca-support-online/product-content/recommended-reading/security-notices/ca20160323-01-security-notice-for-ca-single-sign-on-web-agents.aspx

75 See Xforce report https://exchange.xforce.ibmcloud.com/vulnerabilities/89055

THREAT	WEAKNESS	VULNERABILITY
		access confidential information including backup data, or violate its integrity.[76]
Usurpation	Access Control	Unauthenticated-access and privilege-escalation vulnerability in the Storage Manager GUI for Virtual Environments **Consequence:** this vulnerability allows access and privilege escalation to remote users. An attacker could then maliciously use operating-system commands for actions that seriously impact storage-system operations, alter files, and access information such as credentials stored on this system. CVE-2015-7426: unauthenticated access, privilege-escalation vulnerability in the storage-management system, injection of OS command.[77]

Exploiting vulnerability in the virtual storage can expose systems to the following consequences.

- *Attacking the storage-management console* can lead to disruption and usurpation. In this class of threats, common weaknesses include injection, input validation, and cross-site scripting. Weaknesses are sometimes due to cryptographic Issues.
- *Attacking the storage system* leads primarily to the disclosure of information.

As an example, a complex exploit based on multiple vulnerabilities against Virtual SAN Appliance was carried our using the Metasploit Framework.[78] [79] [80] Some of the exploited vulnerabilities related to login-buffer overflow and command execution (CVE-2012-3282, CVE-2013-2343).

2.5 Impacts and Risks

Risk evaluation is essential to help organizations properly assess and prioritize their vulnerability-management process. A huge effort has been devoted by national bodies and organizations like MITRE[81], NVD[82] and FIRST[83] to evaluating risks of specific vulnerabilities (i.e., Common Vulnerability Scoring System - CVSS) or generic weaknesses (i.e. Common Weakness Scoring System - CWSS).

- CVSS is the standard for scoring CVE. It provides a simple and fairly repeatable method (i.e., different experts typically generate the same score for a specific vulnerability) to specify the characteristics and impact of vulnerabilities, capturing their severity as a numerical score (easy translated into quantitative representations as used by NVD).
- CWSS, like CVSS, provides a mechanism for the prioritizing weaknesses through numeric scoring. CWSS shows high flexibility, since it allows dealing with incomplete information, a rather common situation when the evaluation is carried out on a generic weakness and not on specific product vulnerabilities.

76 See IBM report http://www-01.ibm.com/support/docview.wss?uid=swg21973086
77 See IBM report http://www-01.ibm.com/support/docview.wss?uid=swg21971484
78 See https://www.exploit-db.com/exploits/27555/
79 See https://www.exploit-db.com/exploits/18901/
80 See Metasploit website https://www.metasploit.com
81 See https://www.mitre.org
82 See National Vulnerability Database (NVD) https://nvd.nist.gov
83 See Forum of Incident Response and Security Teams (FIRST) https://www.first.org

The above approaches share the same general methodology for calculating the score associated with vulnerability or weakness risk based on *attack-likelihood evaluation* (e.g., exploitability metrics in CVSS and attack surfaces in CWSS) and *impact estimation* (e.g., Confidentiality/Integrity/Availability impacts in CVSS, and technical and business Impacts in CWSS). The risk score is then computed as *likelihood multiplied by impact*. For instance, vulnerabilities that are easy to exploit but have negligible impact are scored with a low risk.

Both CVSS and CWSS assume the involvement of an expert analyst (or possibly a set thereof) to evaluate likelihood and impact. However, while CVSS requires domain- and implementation-specific knowledge, CWSS, while permitting domain-specific evaluation, also allows generic domain-independent evaluation. For instance, the SANS Top 25 prioritizes generic weaknesses in CWE using CWSS and considering a software-system scenario. It is important to note that vulnerabilities associated with a generic weakness ranked high by SANS Top 25 may be considered less important in a specific domain. For example, this happens when the specific vulnerability (e.g., SQL injection) cannot be directly triggered by an attacker (e.g., no uncontrolled interaction with users) or its impact is bounded (e.g., DB-level protection). Nevertheless, generic ranking of weaknesses can be considered a useful first approximation of risk evaluation. Some effort has been made toward application-scenario-specific weakness prioritizing by applying scoring systems or a variation or simplification of them. As noted before, the OWASP Top Ten defined and ranked the most significant weaknesses in the web-application domain. Their scoring system is based on quantitative likelihood and technical impact, which allows OWASP users to perform partial customization of the ranking in terms of their environment and business or in terms of threat agents' capabilities that influence their likelihood and application- or business-specific impact.

By his side, in order to help to measure impact for the specific context of virtualized environments and mitigate resulting risks, CSA presented a report focused on the practical implementation of best practices for server-virtualization[84].

A possible alternative *methodology* might provide a two-step application-scenario-specific ranking based on CWSS for prioritizing weaknesses and on CVSS for ranking vulnerability. Weakness groups are first prioritized using CWSS,[85] considering a specific application scenario. Each vulnerability in each weakness group is then ranked using the CVSS score associated with each CVE. As an example, let us consider an organization willing to prioritize the effort and the budget in a vulnerability-mitigation process in a service-virtualization scenario that targets a virtual server based on VMware technology. For simplicity, here, we consider the hypervisor component alone. The organization first prioritizes the weakness groups listed in Figure 2-2. Within each group, it then finds the vulnerabilities that affect VMware hypervisor technology. Finally, it ranks them according to CVSS officially provided by the NVD and CVE repositories. This ranked list represents the output of the proposed methodology and is used to mitigate each vulnerability, one-by-one, in the order retrieved, until the effort and budget planned for mitigation are used up. For simplicity, let us focus on the two weaknesses in the CVE repository that show the greatest number of vulnerabilities for VMware components and, for each, consider two specific vulnerabilities, as follows:

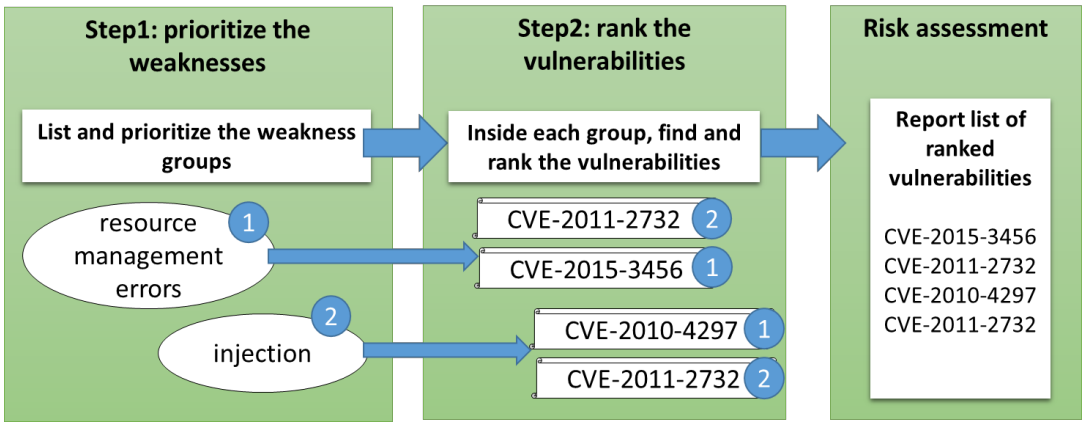
- "resource-management errors" weakness group including CVE-2015-3456 (a vendor-independent weakness in the virtual floppy disk controller) and CVE-2011-2732 (a login-logout weakness in a VMware component)
- "injection" weakness group including CVE-2010-4297 (a command-injection issue in VMware) and CVE-2011-2732 (an injection vulnerability in a VMware component).

84 Best Practices for Mitigating Risks in Virtualized Environments, CSA, April 2015

85 How to compute the CWSS score is available at https://cwe.mitre.org/cwss/cwss_v1.0.1.html

In our example, we assume that "resource-management error" weakness group is ranked higher than the "injection" weakness group. All the CVEs in the "resource-management error" weakness group are then considered of higher importance. Furthermore, among the CVEs in the "resource-management error" weakness group, the CVSS scores show that the most dangerous vulnerability to be mitigated is CVE-2015-3456[86] (VENOM bug). Figure 2-2 shows the ranking obtained in this simple example, which reflects the business-specific peculiarities of the organization.

Figure 2-2 Risk evaluation example using two steps approach



Although there may be other methods to rank vulnerabilities, the approach showed in this section allows a better way to compare vulnerabilities instead of directly comparing CVSS scores. As a matter of fact, CVSS scores are difficult to compare, without considering what weakness they belong to. A high CVSS score for a weakness group that is not important for a specific application scenario should not be considered as bad as a high CVSS score for a crucial weakness group. The approach presented here also allows a better plan for mitigation and builds on a strategy that is weakness-oriented and application-specific. Furthermore, it is in line with the approach followed by the OWASP Top Ten ranking for web-application-specific weaknesses.

In addition, a virtualization environment is subject to rapid evolution due to the automatic, fast patching of components, as well as to modification of the virtualization ecosystem. This would make the risk evaluation outdated. The approach presented here can deal with the peculiarities of virtualization, thanks to the fact that CVE and CWE are rapidly updated in their respective repositories, thus allowing risk evaluation to be periodically rerun (Step 2) to discover new vulnerabilities that need to be taken into account, as well as to re-arrange the priority of intervention to make a virtualized environment more secure. Major changes to the virtualized environment may also change the weakness ranking. Therefore, when they occur the entire risk-evaluation approach needs to be re-evaluated.

In conclusion, risk evaluation provides fundamental grounds for setting up mitigation strategies, thus lowering the impact or the likelihood of an attack. A mitigation strategy based on weaknesses is more generic and more flexible than a mitigation strategy based on vulnerability level. Therefore, instead of complex vulnerability prioritizing, an organization can decide to first prioritize simpler weaknesses, and then implement mitigation based on CVSS score.

86 The CVSS severity for 2015-3456 is ranked 7.7 (HIGH) by NVD.

3. Virtualization Good Practices

A *good/best practice can be generically defined as a proven approach, activity, method or process that produce better results than other approaches.*[87] This section discusses good practices for securing virtualized systems on the basis of components of virtualization and scenarios in section 1. Relevant sources have been collected, reviewed, and mapped to the virtualization weaknesses described in section 2. These sources specify recommendations, controls, safeguards, countermeasures, and good/best practices published by the main institutions or working groups (e.g., ISO, NIST, ISACA), which are fundamental to protecting virtualized components and counteracting the threats in this report. In the editing of this section, we first consider documents produced by the following international bodies:

- International Organization for Standardization (ISO):[88] ISO is an independent, non-governmental international organization with a membership of 163 national standards bodies.
- National Institute of Standards and Technology (NIST):[89] NIST is a measurement standards laboratory, and a non-regulatory agency of the United States Department of Commerce; its mission is to promote innovation and industrial competitiveness.
- Information Systems Audit and Control Association (ISACA):[90] ISACA is an international professional association for information technology management and governance. It proposed the Control Objectives for Information and Related Technologies (COBIT)[91]. As a good-practice framework, COBIT provides an implementable "set of controls over information technology and organizes them around a logical framework of IT-related processes and enablers."

In addition to international bodies, we also consider good practices in virtualized environments proposed in official documents by other not-for-profit organizations, such as the International Information System Security Certification Consortium (ISC²),[92] the Cloud security Alliance (CSA),[93] The MITRE Corporation,[94]

[87] See https://www.axelos.com/Corporate/media/Files/Glossaries/AXELOS-Common-Glossary.pdf
[88] See http://www.iso.org/iso/home.html
[89] See http://www.nist.gov NIST produced two documents specific to virtualization security: i) Karen Scarfone Murugiah Souppaya Paul Hoffman, Guide to Security for Full Virtualization Technologies, Recommendations of the National Institute of Standards and Technology, NIST Special Publication 800-125, January 2011, ii) Ramaswamy Chandramouli, Security Recommendations for Hypervisor Deployment, NIST Special Publication 800-125-A, October 2014.
[90] See http://www.isaca.org/
[91] See http://www.isaca.org/Cobit/pages/default.aspx
[92] See (ISC)² https://www.isc2.org/cissp/default.aspx CISSP certification.
[93] See https://cloudsecurityalliance.org - CSA has also a working group for virtualization security.
[94] See https://www.mitre.org

Computer Associates (CA),[95] Open Network Foundation (ONF),[96] and ICT companies, such as CISCO[97], Symantec,[98] VMware.[99] Finally, we also consider material from relevant scientific books and publications.

Our analysis followed a four-step approach to cover the range of good practices. These have been defined at different levels of abstraction. In the first step, we focused on general-purpose security good practices (section 3.1), including good practices for physical ICT systems that are also relevant for virtualized environments, generic good practices for virtualized environments, and configuration-related good practices considering the central role of component configurations in virtualized environments. In the second step, starting from general-purpose good practices, we present component-specific security good practices for virtualized systems (section 3.2). Each of these targets a single component of virtualization and is very technology specific. In the third step, we provided miscellanea of good practices that target more than one component at time (section 3.3). Finally, in the fourth step, we presented a possible mapping between the identified good practices and weaknesses in section 2 (section 3.4). This mapping permits prioritization of good practices against classes of weaknesses and specific virtualization scenarios, and is summarized in the table in annex B.

During our analysis, we faced two main problems. First, organisations defining good practice do not use a common terminology. For example, NIST proposes *recommendations* or *safeguards*, ISO provides *security controls*, ISACA specifies *best practices that allow bridging the gap between control requirements, technical issues, and business risks*. We therefore constructed an overarching terminology drawing on those already in use, in order to describe good practices in a consistent way. Second, good practices proposed by different organizations are not mutually exclusive. Those from one organisation often either duplicate or overlap those from another. We therefore collapse overlapping/duplicate good practices into one, explicitly providing a reference to the organizations from which the good practice has been derived. In the following, each good practice is described according to the pattern "**CODE DEFINITION [SOURCE ORGANIZATION]"** where

- **CODE** consists of an *acronym* followed by a *number*. The acronym is selected among the following categories of good practices: Physical Layer (PL), Generic (G), Configuration (C), Host/Guest OS (OS), Containers (CON), Hypervisor (HY), Virtual Network (VN), Miscellanea (MISC*)*;
- **DEFINITION** consists of a *title* (optional) and a *description of the good practice*;
- **[SOURCE ORGANIZATION]** lists the *acronyms of the organizations* originally defining the good practice.

To conclude, our analysis shows that publicly available information on virtualization security issues mainly originates from research activities and is based on generic requirements and assumptions, while reports and evidence of real-life experience are not often available. This is mainly due to the fact that security

[95] See http://www.ca.com/us.html and report: Computer Associates, "Virtualization Best Practices", Revision: July, 2010.
[96] See https://www.opennetworking.org/
[97] See http://www.cisco.com
[98] See report: Symantec, "Threats to virtual environment", Version 1.0 (August 2014) in http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/threats_to_virtual_environments.pdf
[99] See report: VMware "Security of the VMware vSphere Hypervisor", January 2014, in http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/whitepaper/techpaper/vmw-white-paper-secrty-vsphr-hyprvsr-uslet-101.pdf

assessment often contains sensitive and critical information, and is managed confidentially for reasons of competitiveness.

3.1 General-purpose security good practices for virtualized environments

A first step in the definition of good practices for virtualized systems and components consists of the identification of general-purpose good practices. These can range from security guidelines for the physical layer beneath the virtualized environments, to generic good practices for component-independent for virtualized environments, and for configuration-related issues. In the following, we discuss general-purpose good practices emerging from documents of main international bodies and not-for-profit organizations.

3.1.1 Physical-layer good practices for virtualized environments

Security good practices for virtualized environments are strongly grounded on security guidelines for the physical layer at their basis.[100] We note that with physical layer we refer to the mix of hardware and software technologies of a generic physical ICT system. For instance, keeping software up-to-date with security patches and all security guidelines at operating system level have a clear impact on the security of virtualized systems. ISO, in its recommendations 27001 and 27002 updated in 2013,[101] proposes general good practices that can prevent unintentional leakages and unauthorized access to sensitive data and systems. In particular, some of them are relevant for both virtualized and physical environments.

- **PL-01 Use of cryptography**. Organizations must define a policy on the use of encryption, plus controls on cryptographic authentication and integrity, such as digital signatures and message authentication codes, and key management.[102] [ISO]
- **PL-02 User awareness** through education and training. This ensures that both general and privileged users understand roles and responsibilities, and act accordingly. [ISO]
- **PL-03 Information classification** with the objective "*to ensure that information receives an appropriate level of protection in accordance with its importance to the organisation*". This general good practice helps identifying the data to be protected. If data are accessed from or transmitted to the cloud, Internet, or another external entity/infrastructure, then the data should be protected according to its classification. [ISO]
- **PL-04 Business requirements of access control**, **user access management**, **and system and application access control** to avoid the common security issue of abuse of authorizations. [ISO]
- **PL-05 Use segregation in networks**. A method of managing the security of large networks is to divide them into separate network domains, based on trust levels along organizational units or some combination. The segregation can be done using different logical networks. [ISO]

Additional security good practices for virtualized environments must consider good practices intended to counteract traditional physical outages. For instance, power supply failure may cause an unexpected power down of the physical system and in turn of the virtual environment deployed on it. If not well managed through, for instance, emergency power system, replication, these types of failures may cause data unavailability, data corruption, or VM data corruption. Finally, an important good practice of fundamental importance for "proper "virtualized environments is to select hardware with the capability of fully

[100] NIST: "*Most existing recommended security practices remain applicable in virtual environments*"
[101] See http://www.iso.org/iso/iso27001 and http://www.iso27001security.com/html/27002.html
[102] ISO 27001 suggests the use of cryptography to deal with unintentional leakages and prevent unauthorized access to sensitive data and systems. However, encryption key management is challenging. Also according to NIST publications, the security for cryptographic keys adds an additional complexity, due to more consumer-provider relationships and the variety of infrastructures "*on which both the key management system and protected resources are located*".

supporting the required virtualization functionalities. To ensure this aim is met, both users and providers should ensure that a list of requirements that hardware must meet (e.g., CPU with hardware virtualization support) is met.

3.1.2 General good practices for virtualized environments

Some generic, component-independent good practices are specifically proposed for virtualized environments. NIST has produced some publications focusing on general recommendations to improve the security of full virtualization technologies.[103] These are also applicable to all virtualization technologies, as follows.

- **G-01: Secure all elements of a full virtualization solution and maintain their security**. The security of a complex virtualization solution depends on the security of each of its components or layers. Organizations must follow standard ICT security guidelines using sound security controls and follow practices such as keeping software up-to-date using host-based firewalls, antivirus, and IDS, to name but a few. [NIST]
- **G-02: Restrict and protect administrative access to the virtualization solution management system.** The security of the entire virtual infrastructure relies on the security of the virtualization management system that controls the hypervisor, supporting operations on guest OSs as well as other administrative actions. Organizations should restrict access to the virtualization management system or any other console interface, supporting hypervisor-level access by authorized administrators only. [NIST]
- **G-03: Ensure that the hypervisor is properly secured.** The security of a hypervisor is crucial and includes actions that are standard for any type of software, such as keeping it up to date by applying security patches. Organizations should follow hypervisors vendor-specific recommendations like disable unused virtual hardware, disable unneeded hypervisor services (e.g., clipboard- or file-sharing), use the hypervisor's monitoring capabilities (i.e., to monitor the security of activities occurring between guest OSs), monitor the hypervisor itself for signs of compromise, provide physical access controls for the hardware on which the hypervisor runs, to name but a few. We note that the latter is mandatory for both hosted and bare-metal hypervisors, where rebooting the computer hosting the hypervisor may allow altering some of the security settings of the hypervisor. [NIST]
- **G-04: Carefully plan the security for a full virtualization solution before installing, configuring, and deploying it**. Planning helps in ensuring security and compliance with all relevant organizational policies. Organizations should consider security from the initial planning stage at the beginning of the systems development life cycle to maximize security and minimize costs. [NIST]

Amongst other issues that needs to be covered is "bring your own devices" and the possibility that they themselves may be part of a cloud infrastructure. As a result, all of the requirements for security are transferred to these devices on the symmetry principle if they connect to cloud services.

NIST and ISACA[104] also present more in-depth studies of security implications of virtualization. In the following, we summarize these studies in terms of additional good practices for virtualization.

---

[103] See Karen Scarfone, Murugiah Souppaya, Paul Hoffman (NIST), "Guide to Security for Full Virtualization Technologies - Recommendations of the National Institute of Standards and Technology", Special Publication 800-125

[104] See ISACA website: http://www.isaca.org/Journal/archives/2011/Volume-1/Pages/Auditing-Security-Risks-in-Virtual-IT-Systems.aspx

- **G-05: Isolate guest OS and perform partitioning**. Hypervisors should allow interactions between VMs only when needed, enabling networking at specific times (e.g., when two VMs have to share storage). Hypervisors should also have policies dealing with physical and logical partitioning. Partitioning allows the preventing of unauthorized access and the reduction of threats of code injection from a VM into another, and decreases the risk of denial of service due to resource exhaustion. In this context, isolation techniques have been deployed to limit i) access to VMs, ii) communications between different VMs, and iii) communications from VMs to hypervisor. VM isolation also helps in mitigating side-channel attacks. [NIST]
- **G-06: Monitor the resources**. The hypervisor or the VMM can be set up to monitor running VMs, network traffic, memory, and processes (introspection). Introspection also provides auditing capabilities and security controls such as firewalling, intrusion detection, and access control. In a typical network configuration, traffic should not be affected by network-based security controls. [NIST]
- **G-07: Properly manage images and snapshots.** Images and snapshots may contain sensitive data, such as passwords and personal data. Proper image management provides significant security and operational benefits to an organization. Images need to be carefully protected against unauthorized access, modification, and replacement by both systems and human actors. A good practice is to keep a small number of known good images of guest OSs that differ, for example, based on the application software that is installed. Snapshots are more risky than images, since they contain the status of the RAM memory. This might include sensitive information that was not even stored on the drive itself, such as passwords in clear text. [NIST]
- **G-08: Perform vulnerability analysis**. Architectural vulnerability analysis provides immediate feedback with respect to the state of system vulnerabilities and makes the architecture more robust and secure. It can be useful to conduct vulnerability analysis by comparing current system attributes to a reference set that consists of valid system samples. We note that vulnerability analysis can be easily conducted following the approach proposed in section 2. [ISACA]
- **G-9: Implement network best practices.** Network best practices should be applied to harden the network interface of the virtual machines. Network segmentation of VMs is suitable to mitigate the risks of various types of network attacks, making network discovery more complex. Physical security devices can be used to keep the trust zones separated. [ISACA]
- **G-10: Prevent single point of failure**. Hypervisor, being a pervasive entity controlling multiple virtual hosts, constitutes a single point of failure. For instance, a replicating malware can rapidly exploit all hypervisors in the networked IT environment. [ISACA]
- **G-11: Control the access to VMs**. Controlled access to virtual environments and proper lockdown of privileges are mandatory to reduce code exploitation through malicious software. [ISACA]
- **G-12: Secure the host OS**. Since the virtualisation layer resides on the host OS, host OS protection is paramount. A compromised host OS may provide a suitable hook for enlarging the compromised perimeter. [ISACA]
- **G-13: Organisational policy for VM security**. At the organizational level, a policy-based security model for hypervisors and host OS should be applied. [ISACA]
- Finally, in the context of cloud computing, CSA[105] highlights the importance of Service Level Agreements (SLAs) and contractual obligations for a proper security management. This good practice is important for any virtualized environments.

[105] See CSA, Security Guidance for Critical Areas of Focus in Cloud Computing v3.0 in https://downloads.cloudsecurityalliance.org/assets/research/security-guidance/csaguide.v3.0.pdf

- **G-14: Define and verify SLA's and contract requirements.** Since there is no physical control over virtualized environments, SLAs and contracts specify requirements that are of paramount importance for risk management. [CSA]
- **G-15**: **Security departments should be involved in the definition of SLAs.** SLA's need to deal with staff competency and suitability, through certification, and, security and criminal record checks [CSCC]. These obligations are symmetric.

3.1.3 Configuration-related good practices for virtualized environments

A virtualized environment is composed of many interacting components, each with a specific configuration. Any misconfiguration in one of these components might open the door to attacks, which might have major disruptive impact. In other words, configuration-associated risks and especially configuration-drift issues can become critical due to the ease of cloning and copying images of VMs. ISACA then specifies configuration-related good practices.

- **C-01: Assess the configuration.** A periodic configuration assessment is needed to achieve a known and trusted state of the virtual environment. [ISACA]
- **C-02: Check the hypervisor configuration.** The integrity of the hypervisor configuration should be checked periodically. [ISACA]
- **C-03: Document properly authorisation changes**. Virtualization permits instant changes to VMs; such changes should be authorised and properly documented. Undetected and unauthorised changes to the VM configuration can introduce security breaches and eventually make the system noncompliant to organisational and regulatory standards. [ISACA]
- **C-04: Configure audit and controls**. Implement a proper configuration audit and control to ensure environmental stability and prevent unexpected threats. Configuration risks can be mitigated by regularly checking the configuration of components against defined standards. ISACA identifies audit as a fundamental practice for security evaluation of virtualized environments. ISACA provides a detailed guideline with a set of prominent audit points[106] divided in several categories: i) how to move from physical to virtual, ii) assess risk, iii) understand the infrastructure and the controls, iv) make a network map of the VM environment, v) evaluate policies, procedures and documentation, vi) evaluate controls, vii) perform network security, viii) encrypt communication, ix) control logical access, x) configure services, xi) configure file sharing between host and guests, xii) configure time synchronisation, xiii) disconnect unused devices, xiv) remote management approaches, xv) patching and vulnerabilities, xvi) collect logs, xvii) make backups, xviii) security from external modifications, xix) denial of service, xx) miscellaneous audit points. [ISACA]
- **C-05: Approved templates for VM deployments**. Templates for VM deployments should be studied and approved before adoption. [ISACA]
- **C-06: Event monitoring.** All events on VMs should be monitored via active-state monitoring of configuration changes to hosts, VMs, clusters, data stores, and virtual networks, to name but a few. [ISACA]
- **C-07: Configuration management database (CMDB)**. A CMDB should be maintained and include information about the location of the images of suspended VMs and the physical-to-virtual mapping. [ISACA]

[106] See Abhik Chaudhuri, von Solms, Dipanwita Chaudhuri, "Auditing Security Risks in Virtual IT Systems" in http://www.isaca.org/Journal/archives/2011/Volume-1/Pages/Auditing-Security-Risks-in-Virtual-IT-Systems.aspx where a list of 141 audit points is provided for virtualized environments.

3.2 Component-specific security good practices for virtualized environments

Starting from general-purpose good practices in the previous section, we provide a detailed grouping on the basis of the specific virtualization components they insist on. The generic cases presented in section 3.1, representing a high-level classification, are further refined in low-level, specific good practices in this section.

3.2.1 Guest OS and Host OS

Guest OS and host OS are basic building blocks of each virtualized environment. Each guest OS and corresponding virtual machine, as well as the host OS and corresponding physical machine, represent (at different levels) the user-space instance where users' data and applications are stored and run. According to general security good practice **G-05**, guest OS requires to be isolated and partitioned to limit the propagation of contagion and the enlargement of the malicious control perimeter, until the worst-case scenario of host OS contagion. Some additional good practices on guest/host OSs are then suggested by CSA[107], [108] in its risk/practices matrix and NIST[109] as follows.

- **OS-01: Protect sensitive data** (passwords, personal data, profiles, history files, encryption keys, license keys, and the like) in the VMs according to the following practices: i) encrypt stored data in VMs, release encryption/decryption keys only to validated and authorized entities, provide options to manage the keys on premises or as a service in the cloud, leverage a policy-based key management system, apply identity and integrity checks when guest OS requests access to storage volumes, ii) develop policies to restrict storage of VM images and snapshots, iii) put policies in place to ensure that backup and failover systems are cleaned when deleting and wiping the VM images (e.g., zero-filling, sanitation) to avoid residual data, iv) consider using cryptographic checksum protection to detect unauthorized changes to images and snapshots, v) identify critical data files within the VM (information classification).[110] [CSA]
- **OS-02: Secure pre-configured/active VMs**. Unauthorized access can lead to VM hardware configuration changes. To mitigate risks, good practices are: i) ensure proper hardening and protection of VM instances through VM guest OS hardening, ii) use VM built-in security measures, leverage third-party security technologies (e.g., discovery and monitoring tools) and provide layered security controls, iii) implement an integrity checksum mechanism for all VM images, iv) encrypt VM images to prevent unauthorized modification, v) implement strict controls around access, creation, and deployment of images/instances, and recording such activity for audit purposes. [CSA]
- **OS-03: Follow the recommended practices for managing the physical OS.** For instance: time synchronization, log management, authentication, remote access, and the like. [NIST]
- **OS-04: Install all updates to the guest OS promptly**. All modern OSs have features that will automatically check for updates and install them.[111] [NIST]
- **OS-05: Back up the virtual drives used by the guest OS on a regular basis, using the same policy for backups as is used for non-virtualized computers in the organization**. [NIST]

---

[107] See CSA "Best Practices for Mitigating Risks in Virtualized Environments" April 2015.
[108] See CSA Cloud Controls Matrix Working Group Cloud Controls Matrix v3.0.1 down load at https://cloudsecurityalliance.org/group/cloud-controls-matrix//
[109] See Karen Scarfone, Murugiah Souppaya, Paul Hoffman, Guide to Security for Full Virtualization Technologies, Recommendations of the National Institute of Standards and Technology, NIST Special Publication 800-125, January 2011 in http://www.nist.gov
[110] There is also a need to have well defined, audited, processes for staff management of this data, including controls over staff physical access to both soft and had data.
[111] Ideally, maintain a test version of OS's that will allow pre-deployment verification of updates.

- **OS-06: Disconnect unused virtual hardware in each guest OS**. Particularly important for virtual drives, but it is also important for virtual network adapters and in general each port. [NIST]
- **OS-07: Use separate authentication solutions for each guest OS, unless there is a particular reason for two guest OSs to share credentials**. [NIST]
- **OS-08: Ensure that virtual devices for the guest OS are associated only with the appropriate physical devices on the host system, such as the mappings between virtual and physical NICs.** [NIST]

If a guest OS on a hosted virtualization system is compromised, NIST suggests two strategies that can be seen as two additional good practices for dealing with compromised system:

- **OS-09: If a guest OS is compromised, assume that all guest OSs on the same hardware have been compromised**. In such a case, revert each guest OS to a known good image that was saved before the compromise. [NIST]. In practice, the compromising event could be specific to a particular OS, in which case, i) only that OS needs to be dealt with on that hardware, ii) all instances of that OS anywhere in the particular cloud may need to dealt with, iii) verify the impact of the compromising event is not evident in other OS's
- **OS-10: Investigate each guest OS for compromise, just as one would during normal scanning for malwares**. If a malware is found, follow the organization's normal security policies. [NIST]

The above good practices are specific to either guest OS, host OS, or both. Host OSs may have some additional good practices coming from "general good practices for virtualized environments" (see section 3.1) due to the fact that, in some configurations, the virtualisation layer resides entirely on the host OS. Among these general good practices, **G-12** (Secure the host OS) must be given high priority. This is essential because the virtualisation layer resides on the host OS, so, host OS protection is paramount. A compromised host OS may provide a suitable hook for enlarging the compromised perimeter) and **G-13** (Organisational policy for VM security. At organizational level, a policy-based security model for hypervisors and host OS should be applied) are of paramount importance.

3.2.2 Containers

Containers, being a special case of guest OSs, share some good practices with guest OSs and take advantage of security measures provided by the host hardware and software. However, containers have some peculiar good practices that depend on the aspects to be secured (e.g., host security, container security, configuration).[112]

- **CON-01: Secure host**. Containers share the same host kernel and depend on the security of the default directory where all related files are stored. The following practices should then be considered: i) create a separate partition for containers, because the directory where all container-related files, including images, are stored might fill up fast and the host could become unusable, ii) update the host, because old kernels lack some of the features required to run containers or have bugs, iii) harden the container host, keeping the host system hardened would ensure that the host vulnerabilities are mitigated, iv) remove all non-essential services from the host, that is, implement only one primary function per server to prevent functions that require different security levels on the same server and avoid mixing various application environments on the same machine, v) only allow trusted users to control the container daemon, vi) ensure that the host running the container

[112] A list of good practises for Docker, but valid for all containers, can be found in the document: Center for Internet Security, "CIS Docker 1.11.0 Benchmark", 2016, available at https://benchmarks.cisecurity.org/tools2/docker/CIS_Docker_1.11.0_Benchmark_v1.0.0.pdf

daemon is running only the essential services, that is, inspect the container host and ensure that it is exclusively used for running containers; examples of other services include web server, database, or any function other than the container's main process.[113]

- **CON-02: Secure containers**. The following practices must be considered: i) stay up to date with container updates, so that vulnerabilities in the software can be mitigated, ii) segregate container groups, iii) drop privileges or run without privileges whenever possible, iv) set volumes to read-only, v) be aware of CPU shares, vi) do not use environment variables to share secrets, vii) set memory limits, viii) do not install unnecessary packages, ix) only run container images from trusted parties, x) make sure the kernel is always updated with the latest security fixes, xi) use a good quality supported host system for running the containers, with regular security updates, xii) do not disable security features of the host operating system, xiii) scan images for security flaws, and xiv) make sure the provider fixes them in a timely manner. [114] [115]
- **CON-03**: **Configure containers properly.** The following practices should be considered: i) only trusted users can control containers daemon, ii) audit daemon for activities and usage, iii) audit container files and directories (daemon runs with 'root' privileges and Its behaviour depends on some key files and directories[116]), iv) restrict network traffic between containers, v) set the logging level, so that (if needed) log events can be reviewed later.

3.2.3 Hypervisor/VMM

The hypervisor is the pillar of virtualization and may represent a single point of failure and vulnerability for all virtualized environments. As a consequence, as also suggested by the general good practices (see **G-03** "Ensure that the hypervisor is properly secured" and **C-02** "Hypervisor configuration checks"), it is a crucial entity to be secured in virtualized environments. Each compromise at hypervisor level comes with high impact (see **G-10** "Prevent single point of failure"), because it permits attackers to take the full control of the virtualized environment including the capability of self-hiding their malicious activities. Good practices targeting hypervisors are tightly connected with the ones about VMM. Often, hypervisor and VMM are used in an interchangeable way. For these reasons, we integrated them in a single section.

NIST and CSA propose a set of security good practices that focus on hypervisor.[117] [118]

- **HY-01: Install all updates to the hypervisor as the vendor releases them.** Most hypervisors have features that will check for updates automatically and install the updates when found. Centralized patch management solutions can also be used to administer updates. [NIST, CSA]
- **HY-02: Restrict administrative access to the management interface of the hypervisor.** All management communication channels should be protected using a dedicated management network or the management network communications should be authenticated and encrypted. [NIST, CSA]
- **HY-03: Synchronize the virtualized infrastructure to a trusted authoritative timeserver.** [NIST, CSA]

---

[113] See Center for Internet Security, CIS Docker 1.11.0 Benchmark (2016)
[114] See Container Solutions (2015) in www.container-solutions.com
[115] Adrian Mouat, Docker Security, Using Containers Safely in Production (2015) with foreword by Dan Walsh (Red Hat)
[116] A list of files and directories can be found in the document "CIS Docker 1.11.0 Benchmark" by Center for Internet Security
[117] Karen Scarfone, Murugiah Souppaya, Paul Hoffman, Guide to Security for Full Virtualization Technologies, Recommendations of the National Institute of Standards and Technology, Special Publication 800-125, January 2011
[118] See CSA "Best Practices for Mitigating Risks in Virtualized Environments", April 2015.

- **HY-04*:* Disconnect unused physical hardware from the host system.** For instance, a removable HDD that might be used for backups should be disconnected when not actively used. Disconnect unused NICs from any networks. [NIST, CSA]
- **HY-05: Disable all hypervisor services such as clipboard- or file-sharing between the guest OS and the host OS unless they are needed.** Each of these services can provide a possible attack hook. File sharing can also be the hook of an attack on systems, where more than one guest OS share the same folder with the host OS. [NIST, CSA]
- **HY-06: Use introspection capabilities to monitor the security of each guest OS.** If a guest OS is compromised, its security controls may be disabled or reconfigured with the intention of supressing any signs of compromise. Ensure security services in the hypervisor permit security monitoring even when the guest OS is compromised. [NIST, CSA]
- **HY-07: Use introspection capabilities to monitor the security of activities occurring between guest OSs.** This is particularly important for those communications that, in a non-virtualized environment, are network-based and monitored by network security controls (e.g., network firewalls, security appliances, and network IDPS sensors). [NIST]
- **HY-08: Carefully monitor the hypervisor for signs of compromise.** This action includes using self-integrity monitoring capabilities that hypervisors may provide, as well as monitoring and analysing hypervisor logs on an on-going basis. [NIST]
- **HY-09: Improve visibility and controls over virtual networks.** Software-based virtual networks created for VM-to-VM communications can hinder security policy enforcement and traffic over virtual networks may not be visible to protection devices (e.g., intrusion detection and prevention systems). The following security controls should be used to mitigate risks: i) monitor SDN and data traffic (similar to physical networks), and determine the tool to use for this task, ii) if separate tools are not installed to monitor communications between VMs, use hypervisor introspection capabilities, iii) implement security technologies that consistently span physical and virtual environments, iv) create consistent configurations and security policies across the physical/virtual network, v) use VM-specific security mechanisms embedded in hypervisor APIs to provide granular traffic monitoring. [CSA]

Both NIST and CSA concentrate on specific good practices for hypervisor deployment. NIST provides *specific good practices for the deployment of virtualization components,*[119] *while* CSA has highlighted some specific risks for the hypervisor in its eleven-entry matrix[120] and proposed to adopt a set of hypervisor security controls for risks mitigation throughout the hypervisor life cycle (development, implementation, provisioning, and management). In the following, since some CSA controls develops on NIST good practices, we present the NIST mapping to the hypervisor's baseline functions.

- **HY-10:** A Type I hypervisor (i.e., directly installed on the host hardware) provides more security assurance than a Type II hypervisor (i.e., running in the host OS as a traditional computer program) due to the reduced attack surface (given the absence of host OS) and the consequent reduced list of vulnerabilities to be addressed.[121] [NIST]

[119] Ramaswamy Chandramouli, Security Recommendations for Hypervisor Deployment, NIST Special Publication 800-125-A, October 2014 available at http://csrc.nist.gov/publications/drafts/800-125a/sp800-125a_draft.pdf
[120] See CSA "Best Practices for Mitigating Risks in Virtualized Environments", April 2015, available at https://downloads.cloudsecurityalliance.org/whitepapers/Best_Practices_for%20_Mitigating_Risks_Virtual_Environments_April2015_4-1-15_GLM5.pdf
[121] See "Security Recommendations for Hypervisor Deployment": Security Recommendation HY-SR-1.

- **HY-11:** A Type I hypervisor platform with hardware-assisted virtualization (both instruction set and memory management) provides greater security assurance than hypervisors with purely software-assisted virtualization. In fact, hardware-assisted virtualization i) lowers the probability of buffer overflow attacks thanks to the hardware support for memory management, ii) provides advanced execution host (root) mode and guest (non-root) mode, so that any exploit code in guest OS cannot undermine the security provided by hypervisor code, iii) provides better safety against VM escape attack (using features such as Direct Memory Access), iv) allows the execution of unmodified guest OSs along with their native device drivers (implying a more robust vulnerability patching). [NIST]
- **HY-12:** The Type I hypervisor should be part of an overall infrastructure that contains: i) hardware supporting a Measured Launch Environment (MLE) and standard Trusted Platform Module (TPM) (TPM), ii) attestation process providing a chain of trust starting from the hardware to all hypervisor components (i.e., assurance that all launched components have not been tampered with and their version is correct). [NIST]
- **HY-13:** A functional hypervisor management console with disk footprint and smaller number of exposed interfaces is easier to be verified, presents a smaller attack surface, and can provide better security assurance. [NIST]
- **HY-14:** The hypervisor should have a boot configuration choice to disallow the use of non-certified drivers and, if the architecture permits, the emulation process should be confined to an unprivileged VM to limit the impact of a faulty device driver code. [NIST]
- **HY-15:** Avoid high memory over-commitment. The ratio of the sum of the RAM assigned to all VMs to the RAM memory of the physical host should never be very high.[122] [NIST]
- **HY-16:** The hypervisor should guarantee physical RAM for every VM along with a limit to this value, and permit the prioritization of the required RAM resource in situations of contention. [NIST]
- **HY-17:** The number of virtual CPUs allocated to each deployed VM should be strictly less than the total number of cores in the hypervisor host. [NIST]
- **HY-18:** The hypervisor should provide features to specify lower and upper bounds for CPU clock cycles allocated to each VM, and a priority score, in order to facilitate scheduling in situations of CPU contention. [NIST]
- **HY-19:** The VM image library should reside outside of the hypervisor host, the library should have strict access control, and each of the images of the library should have a digital signature. [NIST]
- **HY-20:** Mechanisms for security monitoring and security policy enforcement of VM operations (malicious processes and traffic going in/out of VM) should be in place. [NIST]
- **HY-21:** Solutions for the security monitoring and the security policy enforcement of the production VMs should be based “outside of VMs”, should run in a security-hardened VM and should leverage the virtual machine introspection capabilities of the hypervisor.[123] [NIST]
- **HY-22:** The access control solution for VM administration should have granular capabilities (both at permission-assignment and object levels of VMs, or logical grouping of VMs) and the ability to specify “deny” permissions to specific objects. [NIST]
- **HY-23:** The number of users and privileged accounts requiring direct access to hypervisor host should be limited to the bare minimum. [NIST]

[122] NIST suggests a typical ratio of 1.5: 1. For example, if a virtualized host has 64 GB of RAM a maximum 96 GB RAM should be associated with all the running VMs. See also VMware document https://labs.vmware.com/vmtj/memory-overcommitment-in-the-esx-server

[123] This kind of security tools are often referred to as Security Virtual Appliance (SVA): they have access to the introspection APIs of the hypervisor and are independent of the virtual network configuration. Examples are the VMware vShield suite, the Symantec Endpoint Protection Security Virtual Appliance.

- **HY-24:** The user and privileged accounts on the hypervisor must be integrated with the enterprise directory infrastructure (e.g., LDAP, Active Directory). This choice guarantees that authentication is through a robust authentication protocol (e.g., Kerberos), enforces corporate security policies (e.g., password policies), and handles the changes to the user account list (e.g., user deletions). [NIST]
- **HY-25:** The remote access protocol used to access the hypervisor console should have configuration options to deny access and restrict access only to a specified list of accounts. This must include hypervisor root account access. [NIST]
- **HY-26:** Always use hypervisor features enabling i) the definition of a "gold configuration" for a hypervisor deployment, ii) the automated application of the gold configurations to new/existing hypervisor installations, iii) the check of compliance of existing hypervisor installations against the gold configurations. [NIST]
- **HY-27:** A hypervisor patch management practice must be in place. [NIST]
- **HY-28:** Configure the built-in hypervisor firewall to allow only necessary ports and protocols. [NIST]
- **HY-29:** Generate, if possible, logs in a standardized format to help leverage the use of tools with good analytical capabilities. [NIST]
- **HY-30:** Use a dedicated virtual network segment to protect VM management and hypervisor, and enforce traffic controls using firewall (e.g., designate the subnets from which incoming traffic into the management interface is allowed). [NIST]
- **HY-31:** Communications from a given VM to the physical network should be enabled by establishing multiple communication paths within the virtualized host. [NIST]
- **HY-32:** The hypervisor should allow the specification of traffic rate limits to prevent DOS attacks against one virtual server from compromising a complete hypervisor.

3.2.4 Virtual network

Virtual networks pose important challenges to the security of a virtualized environment, where virtual network devices are completely controlled in software and the network protocols/stack simulated to create precise replicas of their physical counter-parts. Virtual networks and corresponding virtual network components are therefore crucial entities, which must be secured to increasing the reliability and trustworthiness of any virtualized environment. Being simulated in software, virtual networks impose fundamental requirements on network isolation and segregation, privileges management, access control, and network administration. Also the separation between the control and data planes introduces new security challenges related to the controllers (which control the entire network) and communications related to the control plane.

According to the general security good practices (see **PL-01** "Use of cryptography" and **PL-05** "Use segregation in networks"), virtual networks require the utilization of cryptographic protocols and the splitting of network into sub-networks. Moreover, some of the good practices identified for hypervisor and VMM in the previous section are relevant for virtual networks as well. In particular, good practices **HY-09** (controls over virtual networks), **HY-24** (integration of users into the enterprise directory infrastructure), **HY-28** (about firewall, ports and network protocols), **HY-30** (network segmentation for management operations) can be applied to virtual networks to guarantee their security.

However, the peculiarities of virtual networks (e.g., centralized control, programmability, cross-domain integration) introduce new security challenges and generate corresponding good practices, a process which

is not yet complete In this context, the Open Network Foundation (ONF) proposes 7 security principles (good practices) for software-defined networks,[124] which are summarised in the following.

- **VN-01: Clearly define security dependencies and trust boundaries.** The specification of security dependencies between components represents a fundamental step in the definition of security mechanisms for virtual networks. Moreover, proper specification of trust boundaries (e.g., based on area of privileges, information flow) supports more precise risk analysis and security evaluation. [ONF]
- **VN-02: Assure robust identity.** Privileges management and access control are paramount to guarantee virtual network security. It is important to implement a strong identity framework, where each device/user can be uniquely identified in a trusted way. A strong identity framework permits to support correct authentication, authorization, and accountability both within the trust boundaries and when external actors are involved. [ONF]
- **VN-03: Build security on open standards.** The adoption of open standards increase portability and interoperability. Existing algorithms and protocols with proven properties are recommended, especially in the security realm, where algorithms and protocols verification is a tedious and time-consuming activity. [ONF]
- **VN-04: Protect operational reference data.** The protection of operational reference data (e.g., credentials, sequence numbers, cryptographic keys) is at the basis of a strong security mechanism and secure virtual network. Incorrect or compromised data may bring to unexpected behaviour or unauthorized access/management. [ONF]
- **VN-05: Make systems secure by default.** Security controls can be configured at different security levels to accomplish different users' requirements. In general, the system should provide minimum security, by deploying and configuring a minimum set of security controls. [ONF]
- **VN-06: Provide accountability and traceability.** Security controls behaviour must be logged to support auditing. Based on auditing activities, relevant sequence of actions can be identified and linked back to the relevant users (see **VN-2**). Security of logged data must be protected. [ONF]
- **VN-07: Properties of manageable security controls.** The introduction of a new security control in a protocol or standard must follow a strict process including clear definition of security objectives, security impact evaluation, support for backward compatibility, to name but a few,. [ONF]

Finally, some suggestions of architectural design, network access control solutions, proposals for incremental deployment of SDNs, commercial applications that detect and resolve DDoS attacks, and generic network monitoring tools are suggested by ENISA[125] in its "Threat Landscape and Good Practice Guide for SDN". The report provides an overview of techniques/tools, including SDN architectures and development frameworks, network access control and troubleshooting solutions, commercial applications for network monitoring, which (partially) address SDN threats.

3.2.5 Virtual storage

A virtualized storage system abstracts a physical storage in a single storage device, which can be accessed either over the network or through a direct connection. Stored data can be only logically partitioned in different storages, while they belong to the same shared storage. This scenario adds a level of complexity

[124] Open Network Foundation (ONF), Principles and Practices for Securing Software-Defined Networks, January 2015, https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/Principles_and_Practices_for_Securing_Software-Defined_Networks_applied_to_OFv1.3.4_V1.0.pdf

[125] See ENISA "Threat Landscape and Good Practice Guide for Software Defined Networks/5G", December 2015, in https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends?tab=publications

with respect to traditional physical storage systems where there is a one-to-one mapping between data owner and physical device. As a consequence, in addition to traditional good practices related to the protection of confidentiality, integrity and availability of data, and protection of sensitive reference data such as keys and credentials, some new challenges and good practices emerge in a virtualized system. First, a new administration layer is added in a virtualized storage (e.g., SAN), where the physical storage administrator is decoupled by the data owner and manager. Second, the sharing of a single physical system introduces the necessity of guaranteeing proper isolation and separation between users. Third, shared media and communication channels require proper protection of data in transit also in a local environment. Fourth, each storage should be subject to strict SLAs and quality of service requirements insisting on best practice performance, utilization policies, availability, and data location (where data are stored, under which privacy policies and jurisdictions should be managed) capabilities.

Since a virtualized storage can be seen as yet another service of a virtualized environment, it should be subject to some of the good practices discussed in this section. In particular, the following general good practices can be applied to the storage systems to guarantee their security:

- The utilization of cryptographic protocols (see **PL-01** "Use of cryptography"),
- The identification of the data to be protected (see **PL-03** "Information classification"),[126]
- A user access management system must be in place (see **PL-04** "Business requirements of access control"),
- The security of all the virtualization sub-component (see **G-01** "Secure all elements of a full virtualization solution and maintain their security"),
- The restriction and the protection of the administrative access (see **G-02"**Restrict and protect administrative access to the virtualization solution management system").

As a further general good practice, a virtual storage system must ensure that:

- All the data of the virtualized solution can be retrieved any time through back-up system and/or disaster recovery facilities (see **OS-05:** "Back up the virtual drives used by the guest OS on a regular basis, using the same policy for backups as is used for non-virtualized computers in the organization" and **G-07**: "Properly manage images and snapshots"; Symantec explicitly suggest the use of a disaster recovery facility[127]). The backup regimes must allow for the delayed detection malware, in which case, the malware exist is backup images created after it infected the host OS. This means that backups made after the arrival of the malware (but before its detection), must be scanned, AND a safe image identified
- SLAs and contracts are properly defined and enforced involving security departments (see **G-14**: "Define and verify SLA's and contract requirements" and **G-15:** "Security departments should be involved in the definition of SLAs").

Moreover, some of the other components' good practices in this section also apply to a virtual storage as discussed in the following. Similarly to the hypervisor, a virtual storage system must:

- Install all updates as the vendor releases them (see **HY-01**), subject to their verification as safe,
- Restrict administrative access to the management interface (see **HY-02**),

[126] CISSP reports two kind of data classification: i) commercial data classification: Sensitive, Confidential, Private, Proprietary, Public, ii) military data classification: Top Secret, Secret, Confidential, Sensitive But Unclassified (SBU), Unclassified.

[127] See Symantec report: Candid Wueest, "Threats to virtual environments", Version 1.0, August 2014

- Synchronize to a trusted authoritative time server (see **HY-03**),
- Monitor for signs of compromise, using self-integrity monitoring capabilities (see HY-08),
- Integrate all the privileged accounts with the enterprise directory infrastructure (e.g., LDAP, Active Directory.) (see **HY-22**).
- Generate, if possible, logs in a standardized format to help leverage the use of tools with good analytical capabilities (see **HY-29**).

Similarly to virtual networks, a virtual storage system must:

- Ensure an identity framework with authentication and authorization functionalities in place (see **VN-02** "Assure robust identity"),
- Protect the confidentiality, integrity and availability of sensitive operational reference data (see **VN-04** "Protect operational reference data"),
- Provide different security levels (see **VN-6** "Make systems secure by default"),
- Provide auditing functionalities (see **VN-7** "Provide accountability and traceability")

Finally, as is the case with the guest/host OS, a virtual storage system must protect passwords, personal data, profiles, history files, encryption keys, license keys, and the like (see **OS-01**: Protect sensitive data).

3.3 Miscellaneous (good practices across different components of virtualization)

Some good practices do apply to a combination of virtualization components rather than to some specific example. In this section, we discuss this, starting from the work done by CSA[128] in highlighting some miscellaneous good practices.

- ***MISC-01:*** **Control the proliferation of VMs**.[129] Since VM instances are easily created and existing instances can be cloned and copied to physical servers and virtual storage, the number of dormant VM disk files increases and security monitoring is much more complex. To mitigate this risk, organizations must consider the following security controls in storage and guest OS: i) manage VM lifecycle with effective policies, guidelines, and processes, ii) use a formal change management process to control creation, storage, and use of VM images, iii) use a small number of known good and timely patched[130] images, this implies verifying that vendor updates are safe, iv) use continuous monitoring to discover dormant virtual systems, and the applications running on them, v) apply security configuration changes to VMs using management/patching solutions, vi) ensure all storage capabilities are properly erased. [CSA]
- ***MISC-02:*** **Secure offline and dormant VMs**.[131] Organizations should provision and decommission VMs in controlled environments, schedule maintenance, and provide disaster recovery facility. Simply powering dormant and offline VMs can deviate from current security baselines and introduce security vulnerabilities. To mitigate risks, organizations must consider the following security controls: i) control backup, archiving, distribution, and restart of VMs with effective policies, guidelines, and processes, ii) use virtualization management solutions to examine, patch, and apply security configuration changes, iii) create a controlled environment to apply security patches and

[128] CSA, Best Practices for Mitigating Risks in Virtualized Environments, April 2015
[129] This is sometimes referred as VM sprawl.
[130] NIST defines them as "gold" images.
[131] The state of a VM can range from active (running), to dormant (suspended), to offline (shut down).

control policies to offline or dormant VMs, iv) design appropriate architecture and monitor virtual appliances that provide critical infrastructure. [CSA]

- ***MISC-03:*** **Do not mix VMs (and corresponding workloads) with different trust levels on the same physical server.** To mitigate risks, organizations must consider the following security controls: i) implement policies and processes to categorize systems and data (security classifications), ii) assign users with workloads of different trust levels to different VLAN networks, iii) run workloads of different trust levels on different physical and logical networks (segregation), iv) use firewalls to isolate groups of VMs from other hosted groups (e.g., production, development, cloud-resident), v) design and implement proper access from each trust level to physical and virtual management and security systems. [CSA]
- ***MISC-04:*** **Risk due to cloud service provider APIs.** In case of a hybrid cloud (i.e., use of private and public cloud infrastructure services at the same time), some services such as enterprise identification, authentication, policy management, and governance frameworks cannot extend into the public cloud. To mitigate risks, organizations must consider the following security controls: i) implement strong authentication and granular access control, ii) use two different authentication zones (one for internal organizational systems and another for external systems), iii) transmit Active Directory/LDAP traffic via a private out-of-band encrypted channel, iv) use identity federation, v) apply enterprise security, compliance, and governance policies to assets managed in hybrid clouds. [CSA]

3.4 Map good practices on weaknesses

Section 2.5 presented a risk-based approach for the prioritization of weaknesses. The prioritization is fundamental for ranking vulnerabilities and organizing a proper mitigation strategy. Although the good practices in this section can be considered as a single means to ensure security in virtualized environments, they can also be prioritized following the weaknesses prioritization.

A possible mapping between the good practices in this section and the weakness groups (produced using the CWE, Common Weakness Enumeration framework[132]) in section 2 is presented in the table in annex B. Recalling that the prioritization of weaknesses depends on the considered scenario, the mapping in the table can drive the selection of good practices according to the specific virtualization scenarios. In other words, following the direct mapping between good practices and weaknesses, also the application of good practices becomes scenario-specific, supporting tailored solutions for securing virtualized systems. As an example, the use of virtualization may facilitate fault injection into non-virtualized systems.[133] To mitigate this specific risk, the administrators could use the good practices in the table proposed for weakness injection.

[132] See CWE Common Weakness Enumeration- A Community Developed Dictionary of Software Weakness Type https://cwe.mitre.org/index.html

[133] Michael Le and Yuval Tamir, Fault Injection in Virtualized Systems – Challenges and Applications, IEEE Transactions on dependable and secure computing, Vol. 12 May/June 2016.

4. Gap Analysis and policy context

In this section, we provide a gap analysis presenting the areas of virtualization threat mitigation, security, and good practices where further research and investigations are needed. For each identified gap, we propose an overview of current on-going activities at European and international level, the possible research opportunities, and, when applicable, the policy context, the legal framework and the most relevant and active initiatives addressing the challenges of security in virtualized technologies.

Our analysis first focuses on gaps related to data collection, management, and protection. In fact, shared virtualization environments expose data to new and increased risks of unauthorized data access and management. In this context, gaps on the use of cryptography are first considered. Cryptography represents a main mitigation/countermeasure for virtualized environments, though it impacts on the performance of the hosting infrastructure and introduces new issues on key management. Gaps on privacy issues related to virtualized environments and data management are also examined. More specifically gaps on privacy include data remanence issues. Proper management of data deletion is paramount due to shared and multi-tenant nature of virtualized environments, where sharing of resources, VM images managing, snapshots and cloning of VM images can create opportunities for privacy violations.

Our analysis then focuses on gaps related to the virtualization infrastructure and components. Among them, gaps on isolation, multi-tenancy, and resource management are analysed. These gaps insist on peculiarities of multi-tenant and multi-layer virtualized infrastructures, and consider approaches for performance management.

Our analysis finally considers post-evaluation and transversal gaps involving gaps on roles and human resources, assurance and monitoring techniques, forensics, and standards. These gaps are introduced by the need for evaluating the behaviour of a virtualized system (including its users and administrators) at both runtime and *a posteriori* (post-execution), to take corrective actions and adapt it to contextual changes. This also introduces pressing requests for standards maximizing interoperability between diverse approaches.

Concluding we present a set of recommendations targeting data owners, administrators, and developers, as well as policy makers and human resources.

4.1 Gaps on the use of cryptography

Gaps related to the use of cryptography in virtualized environments are first related to performance and scalability of all components. As in physical environments, cryptography adds a layer of complexity also in virtualized environments. In other words, the overhead introduced by cryptographic-based security mitigations can affect the availability of the virtualized system and open the door to denial of service attacks. Suitable cryptography must keep performance under control as complexity increases. Performance is in fact a key element of any system as also claimed in the European Union Annual Work Plan 2015[134]: *"due to the complexity of the signalling systems and the differences between sites and applications, a large amount of tests must be carried out on-site. On-site tests take significant effort in terms of time and cost (about 5 to 10 times the effort compared to similar tests done in the lab). The challenge is to reduce on-site tests for signalling systems, leading to reducing overall testing costs."*

[134] See http://ec.europa.eu/research/participants/data/ref/h2020/other/wp/jtis/h2020-wp15-shift2rail_en.pdf

Different kinds of initiatives and research are being undertaken in the context of cryptographic components, as for example in the field of Trusted Computing (TC) with the specification of virtual Trusted Platform Module (vTPM). Virtual TPMs aimed to enable trusted computing for an unlimited number of virtual machines on a single hardware platform but this is still considered less robust than its hardware counterpart.[135]

New approaches to cryptography, such as the notion of "cryptography-as-a-service" in cloud environments[136] and the so called post-quantum cryptography, are also emerging. Post-quantum cryptography refers to algorithms that are strong enough to counteract quantum computer attacks.[137] Many cryptographers are currently working on designing new algorithms and cryptographic primitives that will be applied when quantum-computing will become a threat and quantum adversaries a reality.

The section on Gaps on cryptography also considers those related to key management in virtualized environments. In a virtualized system or, even worse, in a distributed and virtualized environment like the cloud, the problem of safely storing and managing keys is a well-recognised issue and still a research challenge.[138] This gap affects various components, such as hypervisor, guest machines, network and storage.

4.1.1 Overview of current activities

Current and past research at both academic and industrial level is focusing/has focused on the definition of approaches supporting high performance cryptography.[139] [140] [141] [142] The generic challenge of performance and scalability in virtualization environments also led to European research calls in the topic "IT virtualization of testing environment"[143] with the promise to make available faster hypervisors and virtualization components. Several Horizon 2020 European projects are working in this field. **MIcro KErneL virtualizAtioN for hiGh pErfOrmance cLOud and hpc systems** (MiKELANGELO)[144] project proposes a novel and fast hypervisor architecture (called superfast KVM-based hypervisor or sKVM) aimed to improve the I/O performance of virtualised infrastructures and applications. **Software Defined Storage for Big Data**

[135] Jordi Cucurull, Sandra Guasch, Virtual TPM for a secure cloud: fallacy or reality?, RECSI 2014
[136] See proceedings of the International Conference on Applied Cryptography and Network Security conferences, for example Client-controlled Cryptography-as-a-Service in the Cloud (ACNS 2013, see https://www.trust.informatik.tu-darmstadt.de/publications/publication-%20details/?no_cache=1&tx_bibtex_pi1[pub_id]=TUD-CS-2013-0089), or Berson et al. Cryptography as a Network Service in http://www.csl.sri.com/users/ddean/papers/ndss2001b.pdf
[137] According to many crypto-analysts, the security of the currently popular algorithms used in cryptography relies on one of three hard mathematical problems: the integer factorization problem, the discrete logarithm problem or the elliptic-curve discrete logarithm problem. All of these problems can be easily solved on a sufficiently powerful quantum computer. See for example Peter W. Shor "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer" in https://arxiv.org/abs/quant-ph/9508027 and Bernstein "Introduction to post-quantum cryptography" in http://www.springer.com/it/book/9783540887010
[138] See Elaine Barker, William Barker, William Burr, William Polk, and Miles Smid, Recommendation for Key Management, NIST Special Publication 800-57, 2007
[139] See High Performance Lattice Cryptography, HiPerLatCryp Project, http://cordis.europa.eu/project/rcn/96161_en.html
[140] See Apache Commons Crypto, https://commons.apache.org/proper/commons-crypto/project-summary.html
[141] Achieving high performance for Advanced Encryption Standard (AES) applications, http://www.ibm.com/developerworks/library/l-achieving-high-performance-aes/index.html
[142] See HIPS project, ERC consolidator grant, Prof. Lindell, http://crypto.biu.ac.il/hips
[143] See for example the topic identifier S2R-OC-IP2-02-2015 in Horizon 2020 partecipation portal http://ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/topics/s2r-oc-ip2-02-2015.html
[144] See https://www.mikelangelo-project.eu

(IOSTACK)[145] project proposes to enable efficient execution of virtualized analytics applications over virtualized storage resources, to name but a few.

Some European Union funded projects investigated the use of Trusted Computing technology with special emphasis on virtual TPM.[146] **Certification infrastructure for MultiLayer cloUd** (CUMULUS)[147] project proposes to use virtual Trusted Platform Module (vTPM) in the context of certification of cloud services. In the "Cybersecurity and Trustworthy ICT" topic of the ICT-32-2014 call,[148] **empoWering prIvacy and securiTy in non-trusteD envirOnMents** (WITDOM)[149] project focuses on protecting the privacy and security of data by an holistic framework based on cryptography and privacy-by-design paradigm.

However, certification should be extended beyond systems to both operational processes performed by staff, and, the staff themselves. An example of certification of individuals is The International Information Systems Security Certification Consortium [150](ISC)[2]

Several Horizon 2020 European projects are exploring the post-quantum cryptography approach in the "Cybersecurity and Trustworthy ICT" topic of ICT-32-2014 call.[151] For example, **Secure Architectures of Future Emerging Cryptography** (SAFEcrypto)[152] project aims to provide a new generation of practical, robust and physically secure post-quantum cryptographic solutions; **Post-quantum cryptography for long-term security** (PQCRYPTO)[153] project aims to allow users to switch to post-quantum cryptography designing a portfolio of high-security post-quantum public-key systems, and improving the speed of these systems for a broad spectrum of real-world applications.

4.2 Gaps on privacy

New challenges to privacy and data protection have emerged in the last years with the progress of communication and storage technologies. The European Commission has strived to boost the overall level of cyber security and digital privacy in Europe, because evidence of accidents and crimes has become central to undermining consumer confidence in the overall online economy.

The more traditional privacy problem incurring when a user provides its data to a third party is today exacerbated in scenarios, like the cloud, where data owners lose, at least partly, control over the status of their data. A huge interdisciplinary research effort is being devoted to finding methods for creating security, privacy, and transparency in such a challenging distributed and shared scenario. The proposed solutions should aim to empower data owners to maintain control over their data, their distribution and sharing, thus providing verifiable and privacy-enhanced data management. Current privacy gaps and challenges are also summarized in the whitepaper on "Challenges for trustworthy (multi-)Cloud-based services in the Digital Single Market (v3.1)" (January 2016)[154] produced within the Data Protection, Security and Privacy in cloud (DPSP) cluster of European projects.[155] These projects have been funded in the context of H2020 LEIT

[145] See http://www.mpstor.com/news-top/news/175-horizon-2020-iostack-project
[146] See http://cordis.europa.eu/fp7/ict/programme/challenge1_en.html
[147] See http://www.cumulus-project.eu
[148] See http://cordis.europa.eu/programme/rcn/664817_en.html
[149] See http://www.witdom.eu/
[150] See https://www.isc2.org/credentials/default.aspx
[151] See http://cordis.europa.eu/programme/rcn/664817_en.html
[152] See http://www.safecrypto.eu
[153] See https://pqcrypto.eu.org
[154] See https://eucloudclusters.files.wordpress.com/2015/05/dpspcluster-whitepaper-v3-1.pdf
[155] See https://eucloudclusters.wordpress.com/data-protection-security-and-privacy-in-the-cloud/

WP2014-2015 call and address research and innovation in the areas of data protection, security and privacy in the cloud. . On his side, other previous ENISA publications[156] can provide an analysis about different privacy by design strategies and identify specific privacy enhancing technologies that can be useful in virtualization.

A novel challenge to privacy in virtualized environments concerns the "data remanence" problem, which is the residual representation of digital data left on a virtual machine disk even after the VM is deleted and some attempts have been made to erase all the data.[157] Data remanence affects virtualization in laaS cloud models because brand new virtual machines could inherit data stored in templates and make unwilling disclosure of sensitive information. Also, being stored in files, cloning and snapshot of a virtual machine can contain data in the volatile memory at the time it was copied. In other cases a user may maliciously claim a large amount of disk space and then scavenge for sensitive data. Even if various techniques have been developed to eliminate data remanence, such as the simple overwriting or the more secure sanitizing and degaussing, they do not fully fit virtualized environments. In fact, in virtualized environment, virtualized storage is often physically inaccessible (i.e., physical storage destruction is not possible), advanced cloud media systems maintain histories of data throughout all the data's life cycle (as a result of backups, for example), VM instances are controlled by third-parties, and simple operations on VMs, such as cloning, include also data in RAM memory that is typically considered volatile in physical systems.

4.2.1 Overview of current activities

Over the past few years, the European Commission has adopted a series of measures to raise Europe's preparedness to ward off cyber incidents. One example is the Directive on security of network and information systems (NIS Directive). This is the first piece of EU-wide legislation on cybersecurity.[158] Trust and security are at the core of the Digital Single Market Strategy,[159] while the fight against cybercrime is one of the three pillars of the European Agenda on Security.[160] In this scenario, the EU Cybersecurity Strategy, adopted in 2013, outlines the principles that guide the EU action, for example the importance of access to the Internet, and the protection of fundamental rights online. The main objective is the protection of the confidentiality and the security of communications, which is rooted in the fundamental right to the respect of private and family life (including communications), as enshrined in the EU Charter of Fundamental Rights. To help in better management of the privacy problem, the General Data Protection Regulation (GDPR[161], adopted in 2016) and the ePrivacy directive[162], are expecting to become the two important instruments to strengthen fundamental rights in the digital age by giving back the control of personal data to citizens, provide clearer rules on customers' rights to privacy and confidentiality of communications online and simplify the regulatory environment for international business.

---

[156] See https://www.enisa.europa.eu/publications/big-data-protection

[157] B. Al Belooshi K. Salah T. Martin E. Damiani, Experimental Proof: Data Remanence in Cloud VMs, 2015 IEEE 8th International Conference on Cloud Computing

[158] The European Parliament adopted the NIS Directive on 6 July 2016. European Commission Vice-President Andrus Ansip, responsible for the Digital Single Market, and Commissioner Günther H. Oettinger, has issued a statement at this occasion. The Directive entered into force in August 2016. Member States have 21 months to transpose the Directive into their national laws and 6 months more to identify operators of essential services.

[159] See http://ec.europa.eu/priorities/digital-single-market/

[160] See http://europa.eu/rapid/press-release_IP-15-4865_en.htm

[161] See https://en.wikipedia.org/wiki/General_Data_Protection_Regulation

[162] See https://ec.europa.eu/digital-single-market/en/news/eprivacy-directive-commission-launches-public-consultation-kick-start-review

Some Horizon 2020 European projects are then exploring the problem of privacy-aware data management in virtualized environments. In the "Advanced Cloud Infrastructures and Services" topic of ICT-07-2014 call,[163] **Enforceable Security in the Cloud to Uphold Data Ownership** (ESCUDO-CLOUD)[164] project focuses on the definition of practical solutions supporting data owners in maintaining control over their data when using the cloud for data storage, processing, and management, minimizing the sacrifice in terms of functionalities. In the same topic, **A Holistic Data Privacy and Security by Design Platform-as-a Service Framework** (PaaSword)[165] project focuses on the definition of a PaaS framework implementing a holistic data privacy and security by design solution, including encrypted storage and context-aware access control. In the "Cybersecurity, Trustworthy ICT" topic of ICT-32-2014 call,[166] **Trust-aware, REliable and Distributed Information SEcurity in the Cloud** (TREDISEC)[167] project aims to provide a single framework that builds on existing/novel **cryptographic protocols and system security mechanisms**, providing data confidentiality, integrity, and availability guarantees. At the same time, it aims to support efficient storage and data processing across multiple tenants.

Moreover, the Storage Network Industry Association (SNIA)[168] has suggested a set of remedial mechanisms for data remanence problem and policies about object reuse. However, there is no clear standard for recycling of memory and disks, and data encryption is still the more secure option, since it prevents the reconstruction from residual data after erasing.[169] In addition, disk sanitization techniques used for hard drives do not work on Solid State Devices (SSD), given that the internal architecture of an SSD is very different from one of a hard disk drive. Researchers in the industry are developing built-in commands to instruct on-board firmware to run a sanitization standard protocol on the drive to remove all data.

Finally, privacy by compartmentalization (or isolation) is an emerging trend in virtualization solutions for desktop applications (e.g., QbesOS). This paradigm is providing encouraging results in terms of privacy and security using lightweight hypervisors, such as Xen, as a means for providing strong isolation features at application level (e.g., AppVM).

4.3 Gaps on multi tenancy, isolation, and resource management

Virtualized systems are often at the basis of multi-tenant systems (e.g., cloud), where a tenant, that is, a user or a group of users, share a common access to a resource. In a multi-tenant architecture, a resource provides every tenant with a dedicated share, that is, data, configuration, management, functionality and non-functional properties.

Isolation among virtual machines then become paramount and refers to the capability of isolating the behaviour of multiple VMs among each other, despite the fact they share the same physical hardware and physical resources. Although a number of solutions for isolation are actually in place[170] [171], VM isolation in

[163] See http://cordis.europa.eu/programme/rcn/664792_en.html
[164] See http://www.escudocloud.eu/
[165] See https://www.paasword.eu/
[166] See http://cordis.europa.eu/programme/rcn/664817_en.html
[167] See http://www.tredisec.eu/
[168] See http://www.snia.org
[169] See Farzad Sabahi, Secure Virtualization for Cloud Environment Using Hypervisor-based Technology
[170] Rodero-Merino, L., Vaquero, L. M., Caron, E., Muresan, A., & Desprez, F. (2012). Building safe PaaS clouds: A survey on security in multitenant software platforms. computers & security, 31(1), 96-108.
[171] Ochei, Laud Charles, Andrei Petrovski, and Julian M. Bass. "Evaluating degrees of tenant isolation in multitenancy patterns: A case study of cloud-hosted Version Control System (VCS)." Information Society (i-Society), 2015 International Conference on. IEEE, 2015.

virtualized environments is far from being perfect and many forms of attacks are possible, such as covert channel attacks (i.e. secret channels that exist between two supposedly isolated environments), malware attacks and attacks in migration.[172] The major gap to fill is to find a good balance between full isolation (highest security) and the need to control and monitor.

In addition to the gap relating to isolation, in a multi-tenant system, a major gap to be considered is in the context of solutions aimed to control interactions between components of virtualization that are required for a proper functioning of the system. This scenario might open the door to attacks by malicious user controlling a portion of a resource that could try to gain control over other resources. This scenario would lead to denial of service attacks over other users or data leakage.[173]

Finally, an important gap to be considered in multi-tenant systems is on the need of optimized resource management and scaling in response to variable loads caused by different tenants. This can increase utilisation efficiency at a lower operational cost, and guarantee a desired level of Quality of Service (such as the response time) to the end-users, including an increased resilience to attacks to system availability and reliability.

4.3.1 Overview of current activities

New architectures for achieving multi-tenancy securely and efficiently in cloud services are under research, such as hypervisor-level multi-tenant file system storage cloud architecture, virtualization-based multi-tenancy (VMT) architecture,[174] and multilayer management systems.[175] For example, VMT architecture, using KVM as hypervisor, implements multi-tenancy by running multiple virtual interface nodes as guests on the hardware of one physical interface node. In such architecture, a kernel crash that occurs only inside the virtual machine dedicated to the customer does not affect other customers, which run in other guests.

New mechanisms that enforce isolation across virtual machines in specific vendor corporate environments and enable new isolation policies under a variety of configurations and workloads are under development.[176] Also, there are projects that bring isolation to desktop systems.[177] Containerisation and some other Linux systems already allow the setup of different user accounts offering isolated sandboxes, but differently from these approaches, new technologies offer VM and GUI-level isolation, without being based on the same monolithic kernel. For example, the Qubes OS project[178] promises to deliver a brand new operating system that offers privacy and security by *compartmentalization*.[179]

---

[172] See the chapter: A Survey on the Security of Virtual Machines (Jithin and Chandran) in Recent Trends in Computer Networks and Distributed Systems Security, Springer, Volume 420 of the series Communications in Computer and Information Science

[173] For instance, a malicious user owning a virtual machine on a virtualized environment can flood its machines of requests to cause a denial of service to all machines co-existing on the same hardware.

[174] See for example Kurmus,Pletka, Cachin Haas (IBM Research), Gupta (UCSD), "A Comparison of Secure Multi-tenancy Architectures for Filesystem Storage Clouds"

[175] Li-Der Chou et al., The Implementation of Multilayer Virtual Network Management System on NetFPGA, National Central University and National Cheng Kung University, Taiwan in
2011 IEEE 17th International Conference on Parallel and Distributed

[176] For Xen see for example Diwaker Gupta (University of California, San Diego), Ludmila Cherkasova, Rob Gardner, and Amin Vahdat (Hewlett-Packard Laboratories) "Enforcing Performance Isolation Across Virtual Machines in Xen".

[177] Liu, Yanbing, et al. "A behavioral anomaly detection strategy based on time series process portraits for desktop virtualization systems." Cluster Computing 18.2 (2015): 979-988.

[178] See https://www.qubes-os.org

[179] See http://invisiblethingslab.com/resources/2014/Software_compartmentalization_vs_physical_separation.pdf

Some projects that focus on other gaps in this document also touch the problem of isolation in multi-tenant, virtualized systems. In the "Cybersecurity and Trustworthy ICT" topic of ICT-32-2014 call,[180] **Trust-aware, REliable and Distributed Information SEcurity in the Cloud** (TREDISEC)[181] project, which focuses on data confidentiality, integrity, and availability, also aims to implement isolation guarantees between individual user's workloads and integrate them in infrastructures with low impact on their performance and efficiency. **MIcro KErneL virtualizAtioN for hiGh pErfOrmance cLOud and hpc systems** (MiKELANGELO)[182] project that proposes a novel and fast hypervisor architecture to improve the I/O performance of virtualised infrastructures and applications, also considers isolation as a fundamental aspect for the security of virtualized environments. MiKELANGELO aims to reduce effects of side channel attacks, using mechanisms at hypervisor level, and mitigate the effects of sharing physical resources with a malicious VM.

Finally, some Horizon 2020 European projects are working to improve optimization of resources in multi-tenant systems. Project **dEcentralized repositories for traNsparent and efficienT vIrtual maChine opErations** (ENTICE)[183] promises an optimisation at the level of the virtual machine images to improve resource usage, operational costs, elasticity, storage use, and other desired Quality of Service related features. **Framework for SELF-organized NETwork management in virtualized and software defined networks** (SELFNET)[184] project aims to design and implement an autonomic network management framework that has self-organizing capabilities in managing network infrastructures. This is achieved with the automatic detection and mitigation of a range of common network problems that are currently still being manually addressed by network professionals and operators. The project therefore aims to significantly reduce the operational costs and improve user experience.

4.4 Gaps on roles and human resources

In virtualized systems, there is a clear gap associated with the need of having different administration levels. This is especially true when virtual storage or sharing of data/resources are considered. Many critical roles in a virtualized environment are defined, such as system administrators and other privileged users, who have access to corporate data systems and can browse data of different customers. Moreover, administrators of virtualized environments could use their grants to access sensitive information, such as cryptographic key repositories. A gap to be filled by current solutions is to find a balance between the protection of users' security and privacy, and the functionalities provided to administrators of the virtualized environment. This gap should also consider the hierarchical approach to system administration, where at the bottom there are the administrators of the physical platform, and on top of them a multi-layered management systems, with all the administrators of the virtual components. Efforts need to be made to develop adaptive management systems, controlling the activities of system administrators.[185]

Another gap to be filled in the near future is on human resources, where lack of skilled personnel with the ability of deploying/configuring a virtualized environment and managing its security, could place virtualized systems at high risks. Different profiles are needed to day including system engineers, system developers, security experts, to name but a few.

[180] See http://cordis.europa.eu/programme/rcn/664817_en.html
[181] See http://www.tredisec.eu/
[182] See https://www.mikelangelo-project.eu
[183] See http://www.entice-project.eu
[184] See http://cordis.europa.eu/project/rcn/197349_en.html
[185] See for example Chou, Yang, Chang, Hong (Taoyuan, Taiwan), "The Implementation of Multilayer Virtual Network Management System on NetFPGA", IEEE 17th International Conference on Parallel and Distributed Systems

In summary, both operational processes performed by staff, and, the staff themselves must be scrutinized to ensure risks are known and controlled. These issues have been mentioned elsewhere in this document.[186]

4.4.1 Overview of current activities

Different research projects and almost all virtualization providers are focusing/focused on providing enhanced access control functionalities, which support the process of managing roles and permissions. More recent solutions focus on Chipertext-Policy Attribute-Based Encryption (CP-ABE),[187] [188] [189] a system for realizing complex access control on encrypted data. With CP-ABE encrypted data can be kept confidential even if the storage server is untrusted. Finally, different deployment paradigms could be of help in guaranteeing the confidentiality of data from the prying eyes of privileged users. For instance, in a hybrid cloud deployment model, some data can be kept in the public infrastructure, some others can be held in a private support.

The keys to closing all gaps concerning human resources lies in human resources awareness, education, and training. Some new online educational web sites are offering specialised courses in the field, for example MIT.[190] Also MOOC websites like Coursera,[191] Udacity,[192] and EdX[193] are available. However, as with all the ICT security, it will take years to fulfil industry's requirements on skilled and trained personnel.

4.5 Gaps on security assurance and SLAs

Security assurance aims to increase the confidence of the users that an infrastructure and its services behave as expected.[194] Assurance can be defined as "*the way to gain justifiable confidence that infrastructure and/or applications will consistently demonstrate one or more security properties, and operationally behave as expected despite failures and attacks*".[195] Among assurance techniques, audit, compliance and certification stand out. However, the intrinsic peculiarities of virtualized systems in general, and cloud systems in particular, make existing techniques almost inapplicable.

A security assurance technique for virtualized systems should consider the intrinsic dynamics of such systems, with their multi-layer architecture made of distributed components, and must depart from the traditional assumption of having a single entity responsible for the whole process. We note that the latter assumption is hardly applicable, since assurance verification in virtualized systems is a continuous and adaptive process.

---

[186] See beginning of section 4.2

[187] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-Policy Attribute-Based Encryption, in Proc. of the IEEE symposium on security and privacy (SP 2007), 2007

[188] K. Yang, X. Jia, K. Ren, and B. Zhang. April 2013. DAC-MACS: Effective data access control for multiauthority cloud storage systems. In Proc. of IEEE INFOCOM 2013

[189] Z. Wan, J. Liu, and R.-H. Deng. 2012. HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing. IEEE TIFS 7, 2 (April 2012), 743–754

[190] See http://ocw.mit.edu/courses/

[191] See https://www.coursera.org

[192] See https://www.udacity.com

[193] See https://www.edx.org

[194] See C.A. Ardagna, R. Asal, E. Damiani, Q.H. Vu, "From Security to Assurance in the Cloud: A Survey," in ACM Computing Surveys (CSUR), 48(1), 2:1-2:50, August 2015

[195] See IATAC and DACS. 2007. Software Security Assurance: State of the Art Report (SOAR). http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA472363.

Current gaps and challenges on assurance are also discussed in the whitepaper on "Challenges for trustworthy (multi-)Cloud-based services in the Digital Single Market (v3.1)" (January 2016)[196] produced by the Data Protection, Security and Privacy in cloud (DPSP) cluster of European projects.[197] In particular, gaps on continuous control of security and privacy conditions, SLA management, and cloud security certification are reported.

Finally, another class of assurance techniques is based on Service Level Agreements (SLAs).[198] SLA-based techniques aim to establish contracts between clients and service providers regulating their interactions, and modelling their expectations in terms of both functional and non-functional agreements.[194] Virtualization infrastructures, due to their multi-tenant nature, introduce gaps in both the definition and enforcement of SLAs. Several properties reported in SLAs, such as performance, suffers from the fact that the virtualized environment is shared among different users asking for different SLAs. In addition, different SLAs can interfere, such as for instance, cryptographic-based security and performance, introducing the need for a solution able to find the best compromise that balances the level of satisfaction of the users.[199] Considering a cloud environment, sharing is just one of the issues that may affect an SLA. Current SLAs, mainly referring to physical systems, should take care of the virtualization peculiarities and virtualized context, as well as their intrinsic dynamics and event-based management, providing continuous verification and negotiation.

4.5.1 Overview of current activities

In the last few years, some approaches started to consider the problem of applying security assurance techniques in virtualized systems. These approaches aim to provide solutions that accomplish the dynamics and diversity of virtualized systems, reducing the impact in terms of resource consumption and security concerns. **Certification infrastructure for MultiLayer cloUd** (CUMULUS)[200] project proposed a certification process and infrastructure for the cloud. Within CUMULUS, a hybrid, multi-layer, and continuous certification process have been provided and integrated with advanced cryptographic components.[201] [202] [203] In addition, other approaches provided the capability of observing virtualized system behaviour and evaluating its compliance with policies and regulations such as, Payment Card Industry Data Security Standard (PCI DSS).[204] [205] [206] Recent Horizon 2020 research calls intercepted the need of assurance in the

[196] See https://eucloudclusters.files.wordpress.com/2015/05/dpspcluster-whitepaper-v3-1.pdf
[197] See https://eucloudclusters.wordpress.com/data-protection-security-and-privacy-in-the-cloud/
[198] P. Wieder, J.M. Butler, W. Theilmann, R. Yahyapour, Service Level Agreements for Cloud Computing, Springer Science & Business Media, 2011
[199] A. Arman, S. Foresti, G. Livraga, P. Samarati, "A Consensus-based Approach for Selecting Cloud Plans," in *Proc. of the 2nd International Forum on Research and Technologies for Society and Industry (RTSI 2016)*, Bologna, Italy, September 7-9, 2016
[200] See http://www.cumulus-project.eu/
[201] See M. Anisetti, C.A. Ardagna, E. Damiani, and F. Saonara. 2013b. A Test-based Security Certification Scheme for Web Services. ACM TWEB 7(2):1–41, May 2013.
[202] See Marco Anisetti, Claudio Agostino Ardagna, Filippo Gaudenzi, Ernesto Damiani, A certification framework for cloud-based services. Proc. of SAC 2016, Pisa, Italy, April 2016
[203] See A. Munoz and A. Mana. June 2013. Bridging the GAP between Software Certification and Trusted Computing for Securing Cloud Computing. In Proc. of IEEE SERVICES 2013, June-July 2013 Santa Clara, CA, USA.
[204] See S. Pearson. 2011. Toward Accountability in the Cloud. IEEE Internet Computing 15, 4, 64–69, 2011
[205] See B. Wang, B. Li, and H. Li. 2014. Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud. IEEE TCC, 2014
[206] See CSA. CloudAudit: Automated Audit, Assertion, Assessment, and Assurance, 2016, https://cloudsecurityalliance.org/research/cloudaudit/

topic "*assurance and certification for trustworthy and secure ICT system, services and components*" [207]. In particular, "DS-01-2016 assurance and certification for trustworthy and secure ICT systems, services and components for both physical and virtual environments" calls for solutions aimed to discover vulnerabilities in both physical and virtual environments, addressing security, reliability and safety assurance at individual phases of the ICT Systems Development Lifecycle. Also, some standardization activities are going on in the context of assurance solutions for the cloud. The CEN/CENELEC WS Requirements and recommendations for assurance in the Cloud (RACS)[208] is producing an overview of current regulatory and standardisation efforts in the context of monitoring and certification of cloud computing services, focusing both on ICT technical specifications and best practice. An example which examines both personnel and system certification is CSCC's Cloud Security Standards: What to Expect and What to Negotiate Version 2.0 (2016)[209]. On the personnel side, issues such as Identity & Access Management are discussed, and stresses data privacy issues as they relate to personnel a well as systems. Mandatory reporting of security breaches is also advocated by CSCC (ibid), and, in the related context of control of government, agencies access to citizens' electronic communications.[210]

Similarly, monitoring and security tools and techniques need to be adapted to monitor malicious activities in distributed, virtualized systems. Traditional Intrusion Detection Systems (IDSs) and Intrusion Prevention Systems (IPSs) may not integrate well into or properly operate within virtualized environments as they did in traditional corporate infrastructures.[211] Host-based systems instead still function on virtual machines, but they now tend to drain shared resources, making installation of security agents more problematic. Many vendors have already adapted their existing IDS and IPS platforms to be more easily integrated with virtualization suites, while some specialized virtualization-specific products are becoming available on the market today.[212] Further efforts are also conducted to allow IDS/IPS to monitor more granular traffic.

Since access control (AC) plays a fundamental role in securing controlled delivery of data services (e.g., workflow management, enterprise calendar, records management) to its users, there are some on-going activities to accommodate all these functions in a single underlying AC framework. For example the NIST Cloud Computing and Virtualization (a sub-group of the "Systems and Emerging Technologies Security Research" (SETS) group) has been designing an AC framework called Policy Machine (PM).[213] PM has then evolved beyond the concept to a prototype implementation.[214]

Finally, some work is ongoing in the context of SLA management. **Making Cloud SLAs readily usable in the EU private sector** (SLA-Ready)[215] project aims to propose a framework providing a common understanding of Cloud services SLAs, increasing standardisation and transparency. The project will support SMEs in making the right decision on what services to use and trust. **SLALOM** project[216] aims to provide a simple,

[207] See http://ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/topics/ds-01-2016.html
[208] See https://www.cen.eu/work/areas/ICT/eBusiness/Pages/WS-RACS.aspx
[209] See http://www.cloud-council.org/deliverables/CSCC-Cloud-Security-Standards-What-to-Expect-and-What-to-Negotiate.pdf
[210] Reed, K Computer scientist calls for urgent 'three-prong action' to control State Internet surveillance Karl Reed La Trobe University, 8/11/2013
[211] See C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan. A survey of intrusion detection techniques in Cloud. Journal of Network and Computer Applications 36, 1, 42–57, June 2013
[212] See blogs in http://www.techtarget.com/network for example http://www.techtarget.com/contributor/Dave-Shackleford
[213] See http://csrc.nist.gov/pm/
[214] The project is available for download in Github, see https://github.com/PM-Master/PM/
[215] See http://www.sla-ready.eu/about-sla-ready
[216] See http://slalom-project.eu/

fair and transparent solution for providers and business users to negotiate SLAs supporting business in the Cloud. Procurement Innovation for Cloud Services in Europe (PICSE)[217] project focuses on the problem of cloud service procurement for the public sector.

4.6 Gaps on forensics

Computer forensics is a branch of digital forensics pertaining to evidence found in digital devices. It is used to conduct investigations into computer related incidents, whether the incident is an external intrusion into your system, internal fraud, or staff breaching your security policy.[218]

Computer forensics in virtualized cloud environments can be extremely challenging, since the traditional approach to evidence collection and recovery is no longer practical. Typically, digital forensics has different steps, such as the identification, the recovery, and the forensic preservation of the evidence, the analysis of the collected material, and finally the presentation of facts and opinions. However, the amount of data to analyse in a virtualized environment, such as the cloud, could be overwhelming and the identification, recovery, and preservation activities could be difficult to implement, due to the fact that data reside remotely and technical operations and controls are increasingly dynamic (both in terms of ownership and management). Furthermore, in a cloud scenario, the jurisdiction can introduce additional problems that must be dealt with: while data in a physical computer can be simply seized, data in the cloud could be distributed across several countries each having different laws and regulations. Finally, in a virtualized environment, it is difficult to keep the original "crime-scene", because the environment and resources are shared between different tenants and therefore the activity of a different tenant can permanently compromise the evidence.

Effective forensics depends in part on timely notification of breaches, either by human actors or by computer based agents. The importance of prompt notification is mentioned as part of the "mandatory" reporting of breaches recommended by CSCC (see[219]), which goes further, and advocates vetting of personnel. [220]

As a general remarks, forensics in virtualized environments requires deeper technical competences that classical forensics in physical systems, as well as a relevant support by the service provider that requires forensics analysis. SMEs that directly benefit from virtualization rarely have internal competences for forensics even just for providing forensics evidence. This gap in providing better forensics evidence is partially covered by assurances approaches such as the ones mentioned in Section 4.5, which indirectly provide evidence usable for forensics analysis, though they are still not fully supported. Some approaches for forensics as a service[221] can also be a viable solution to reduce the amount of competences required to SMEs, though they do not fully address this problem.

4.6.1 Overview of current activities

Data moved to the cloud can be physically stored in different countries, which are different from the country where the data owner resides; for this reason there is an issue in identifying the jurisdiction that applies. For instance, different privacy rules must be enforced in the country where data are stored, while organizations must comply with rules in their own country. In addition, organizations are subject to data privacy laws and

[217] See www.picse.eu
[218] See https://www.sans.org
[219] See http://www.cloud-council.org/deliverables/CSCC-Cloud-Security-Standards-What-to-Expect-and-What-to-Negotiate.pdf, p. 25
[220] (ibid p.26)
[221] Wen, Yuanfeng, et al. "Forensics-as-a-service (faas): computer forensic workflow management and processing using cloud." Cloud computing (2013): 208-214.

Safe Harbour[222] policies that require them to protect and in some cases guarantee data sovereignty. The EU Data Protection Directive (1995)[223] and the new EU General Data Protection Regulation (2016)[224] are the legal instruments that deal with this issue.

Computer forensics need then to consider the fact that often law enforcement officers need to retrieve remote files or access social network contents posted by a suspect to virtualized cloud environments. The seizure of a physical smartphone could open all the owner's virtual environments through the identification tokens stored in the device itself, providing access to cloud virtual storage and social network accounts. Some commercial tools allow doing so[225] and, in this case, the analysis of the evidence can be performed online without the need to download a vast amount of data from the cloud. However, in case of storage encryption, special ad-hoc software must be developed from scratch.[226]

Furthermore, NIST is running the NIST Cloud Computing Forensic Science project. The long-term goal of this project is "*to advance technology, standards, and measurements for cloud computing forensic science that will aid further innovation, as well as lead to increased adoption in both government and industry. NIST aims to contribute towards improved accuracy, reliability, scientific validity, and usefulness of cloud forensic science.*"[227]

In general, this lack of jurisdiction and precise boundaries of a scene of crime call for solutions balancing between the need of investigation and the protection of the privacy of the users. Gaps on privacy and gaps on forensics, as well as corresponding activities in these fields, are therefore strictly intertwined.

4.7 Gaps on standards

Standards about protocols and solutions are key in any field of computer science to foster interoperability, portability, security, performance, and adoption. This is even more important in virtualization, where the diversity of the environment causes the proliferation of ad hoc security solutions that target a small part of the environment. Many EU documents and directives for ICT security have been/are mentioned in EU Horizon 2020 calls for research and innovation projects, to boost standardization efforts. As a consequence, many funded projects in H2020 LEIT WP2014-2015 call addressing research and innovation in the context of data protection, security and privacy in the cloud, also focus on standardization issues. These projects are part of Data Protection, Security and Privacy in cloud (DPSP) cluster, which produced a whitepaper on "Challenges for trustworthy (multi-)Cloud-based services in the Digital Single Market (v3.1)" (January 2016)[228] discussing, among the others, gaps on standardization. In particular, the document describes that: "*there are more than 20 organisations active in standardisation, and virtually hundreds of standards published governing all kinds of aspects relevant for cloud computing. It seems that the current unclear*

[222] A safe harbor is a regulation that specifies that certain conduct will be deemed not to violate a given rule.
[223] Directive 95/46/EC of the European Parliament and of the Council of 24 October (1995) on the protection of individuals with regard to the processing of personal data and on the free movement of such data [1995] OJ L 281/31 (also referred as 'EU Data Protection Directive').
[224] The General Data Protection Regulation (GDPR), adopted in April 2016 (EU 2016/679), is a regulation by which the European Commission intends to strengthen and unify data protection for individuals within the European Union and addresses the export of personal data outside the EU.
[225] See mobile forensic solutions of private companies such as Cellbrite (http://www.cellebrite.com), Oxygen (http://www.oxygen-forensic.com/en/), etc.
[226] For example, in the well-known FBI versus Apple case about the San Bernardino shooting (December 2015), a special software component had to be developed to bypass the devices' security and unlock the phones at the cost of more than one million USD, see http://www.reuters.com/article/us-apple-encryption-fbi-idUSKCN0XI2IB
[227] See https://www.nist.gov/programs-projects/nist-cloud-computing-forensic-science
[228] See https://eucloudclusters.files.wordpress.com/2015/05/dpspcluster-whitepaper-v3-1.pdf

situation is voluntarily induced by major market players to foster incompatibilities and customer lock-in. Although "Cutting through the Jungle of Standards" is defined "Key Action 1" of the European cloud computing strategy, and specific actions for the resolution of the situation were implemented (and are on-going), there is no remedy for the situation expected in the nearer future." Moreover, it identifies gaps on standardization in cloud environments including i) the need of interoperability solutions for implementing standardized services, ii) standard certificates of CSP allowing automatic comparison and selection of offerings, and iii) standardised and transparency in Cloud SLAs. Interoperability of data formats and interface of cloud services is fundamental to ensure compatibility between independent systems. Standardization is one of the prominent way to obtain such interoperability. The adoption of standardised SLAs is a critical step towards easier comparison of the CSPs' cloud offerings (as discussed in Section 4.5), filling in the relative lack of balance between the risks and responsibilities of the customer and the CSP and the technical lack of contextual SLA.

4.7.1 Overview of current activities

In the past, many standardization processes have been conducted to rationalize the adoption and management of virtualized environments with particular attention to the cloud. For example the Cloud Standards Wiki,[229] maintained by the Cloud Standards Customer Council (CSCC[230]), collates the efforts done by its members[231] aimed at standardizing the cloud. Among them, The European Telecommunications Standards Institute (ETSI)[232] has been appointed by the European Commission to coordinate the stakeholders in the cloud standards ecosystems and devise roadmaps about standards in support of EU policy in critical areas such as security, interoperability, data portability, and reversibility. The European Commission has in fact released an European Commission Communication on Cloud Computing titled "Unleashing the Potential of Cloud Computing in Europe,"[233] identifying the capability of cutting through the jungle of standards as one of the key actions to foster mass adoption of cloud computing.

Among the activities of the Cloud Standards Customer Council, we recall the adoption of Open Virtualization Format (OVF),[234] a specification that describes an open, secure, portable, efficient and extensible format for the packaging and distribution of software to be run in VMs, and the Open Cloud Standards Incubator that focuses on standardizing interactions between cloud environments.

Another European initiative in standardization is the European Committee for Standardization (CEN)[235], which provides a platform for the development of European Standards and other technical documents in relation to various kinds of products, materials, services and processes. The organization has delivered and is delivering some documents relating to requirements and recommendations for assurance in the Cloud (RACS).

[229] See http://cloud-standards.org/
[230] http://www.cloud-council.org/
[231] Founding members include IBM, Kaavo, CA Technologies, Rackspace & Software AG, while more than 500 of the world's leading organizations have already joined, including Lockheed Martin, Citigroup, Boeing, State Street Bank, Aetna, AARP, AT&T, Ford Motor Company, Lowe's, and others.
[232] See http://www.etsi.org
[233] See http://ec.europa.eu/transparency/regdoc/rep/1/2012/EN/1-2012-529-EN-F1-1.Pdf
[234] See ANSI INCITS 469 2010 and ISO/IEC DIS 17203
[235] From the French "Comité Européen de Normalisation". It is an association that brings together the National Standardization Bodies of 33 European countries

Finally, ISO/IEC is working on the standard "Information technology -- Cloud computing -- Service level agreement (SLA) framework",[236] which is composed of three main documents focusing on: *i)* overview and concepts, *ii)* metric model, *iii)* core conformance requirements. This standard complements the many other standards proposed by ISO/IEC on cloud and virtualization focusing on assurance and SLAs.

[236] See http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=67545, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=67546, http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=67547

5. Conclusions and Recommendations

This report provided an analysis of the status of virtualization security, discussing threat, countermeasures, best practices, and current gaps. It started from the identification of virtualization components/technologies and described the main application scenarios where virtualization is adopted. Based on this classification, a description of a virtualization-specific threat taxonomy and a view of the Common Weakness Enumeration (CWE) weakness groups that fit virtualization scenarios have been provided. The latter view included weaknesses with published vulnerabilities relevant to virtualization, with special emphasis on those that affect any of the virtualization components identified in the taxonomy of threats. Starting from the weaknesses groups, we presented a description of i) how a virtualized environment exacerbates the weaknesses peculiarities with concrete examples of vulnerabilities taken from the Common Vulnerability Exposure (CVE), ii) for each virtualization component, the main vulnerabilities and corresponding consequences, and, when available, examples of real cyberattacks, iii) a concluding discussion on a possible approach to virtualization-specific, risk-aware prioritization of vulnerabilities. Thereafter the identified threat taxonomy, weaknesses, and vulnerabilities of interest for virtualization, as well as a selection of generic and virtualization component-specific good practices for securing the virtualized environment have been given. Finally, we provided a gap analysis by offering a comparison between identified virtualization threats and identified virtualization countermeasures. The gap analysis presented the areas of virtualization threat mitigation, virtualization security, and good practices where further research and investigation are needed. For each identified gap, we showed an overview of current on-going activities at the European and international levels, possible research opportunities, and, when applicable, the policy context, the legal framework and the most relevant and active initiatives addressing the security challenges for virtualized technologies.

To conclude this report, we provide a set of recommendations for next-generation security in virtualized environments. Since virtualization is today a core enabling technology, recommendations apply more to scenarios such as cloud computing or variations between in-house virtualized infrastructures and cloud-based virtualized infrastructures,[237] than to virtualization itself. [238] The set of identified recommendations can be classified as *general recommendations*, *technical recommendation*, *organizational recommendations*, and *recommendation on human resources*.

The general recommendations target the main stakeholders of a virtualized environment, such as data owners and policy makers. The technical recommendations target owners and administrators of the virtual infrastructures, and developers of corresponding products. The organizational recommendations target owners and administrators of virtual infrastructures, and policy makers as well. The recommendations on human resources target the users of virtual environment assets, such as engineers and technical staff of corporations, small/medium private companies, governmental bodies, as well as final users.

General recommendations (for data owners and policy makers)

[237] See Ramaswamy Chandramouli (NIST), Security Control Variations Between Inhouse and Cloudbased Virtualized Infrastructures, in http://www.nist.gov/customcf/get_pdf.cfm?pub_id=911036

[238] According to NIST virtualization becomes cloud at a certain point of scaling: "*When multiple server virtualization is used for running servers on many hosts and for moving servers from host to host based on changing resource needs, it can be called cloud computing*." See Karen Scarfone, Murugiah Souppaya, Paul Hoffman, Guide to Security for Full Virtualization Technologies, Recommendations of the National Institute of Standards and Technology.

Stakeholders, such as data owners, should assume that a virtualized environment is much more than a traditional environment moved from physical to virtual, since new technological layers are added to the systems. For this reason risk assessment should consider and address all characteristics of a virtual environment, and evaluate i) the current level of security by understanding which virtual components are covered and which are not covered by existing security measures, ii) the effectiveness of the application of good practices adapted from traditional security and privacy tools and techniques. Policy makers should prepare roadmaps comprising security risks with security guidance requirements, interoperability opportunities, portability standards, and technology requirements to bypass the barriers to a broader adoption of virtualization products. Policy makers should also encourage the set-up of working groups specialized in this arena. Policy makers should also foster the definition of clear and ad hoc standards that accomplish the nature of virtualized systems. A revision of policies is necessary to eliminate deficiencies when assurance evaluation, SLA enforcement, and a posteriori forensics analysis are considered. The multi-tenant nature and the unavoidable interference between tenant activities in a virtualized environment must be properly regulated and managed to ensure proper monitoring and evaluation of system operations.

Technical recommendations (for owners, administrators and developers of virtual infrastructures)

Virtual environments introduce new threats, risks, challenges, and also new assets and components. As a consequence, new products are needed to provide effective countermeasures and increase the trustworthiness of such environments. When adopted, these new products must be put in the life cycle only after a careful evaluation, through pilots, aimed to verify and prove their correct behaviour. Often, the best and most successful security products come from third-party vendors committed to apply cutting edge security measures and stay focused on any updates. Also developers of new products may benefit from new tools especially those providing security and privacy functionalities by default. More specifically, provided security tools need to fit virtualization peculiarities and not to be only adaptations of existing techniques. Virtualization components should easily integrate with such tools through open interfaces and APIs. These tools should capture the many steps of security management, including among the others, risk evaluation, security prevention and detection, assurance evaluation and SLA management. In addition,

ENISA has been working in the field of privacy technologies over the last years, providing an inventory of existing privacy-by-design approaches, strategies, and technical building blocks of various degrees of maturity and producing different reports. For example, ENISA published the Privacy and Data Protection by Design report,[239] and other reports more specific to certain scenarios,[240] aimed at analysing privacy-by-design strategies and tools. International bodies are also invited to support the shift to virtual environment-specific security and privacy solutions by implementing a gap analysis on standards, and new standardization activities according to the gaps identified in this report. European projects (e.g., in the context of DPSP cluster[241]) are finally working to the aim of providing new security and privacy solutions for the cloud.

Good practices need to be continuously published and updated involving the participation of standardization bodies and EU authorities, and providing guidelines for the assessment of security tools and components. Independent assessment by third-party experts and authorities should evaluate maturity and correctness of tools, according to these practices.

[239] See https://www.enisa.europa.eu/publications/privacy-and-data-protection-by-design
[240] For example for the Big Data cloud scenario, see https://www.enisa.europa.eu/publications/big-data-protection
[241] See https://eucloudclusters.wordpress.com/data-protection-security-and-privacy-in-the-cloud/

Recommendations for security (in the cloud or elsewhere) often seem to be written as obligations on the supplier carrying penalties, however, as CSCC point out[242](but do really explain), these obligations are symmetric, that is, there are elements of the security requirements which, if not adopted by a client, become the weakest link in the security enforcement chain.

Organizational recommendations (for owners and administrators of virtual infrastructures, and policy makers)

The main organizational recommendation is that everyone has to rethink the corporate organizational chart and ensure that each one is properly staffed to deal with these new technologies. Virtualization is not only the simple use of new software components, but also the development and administration of complex processes and services involving cryptography, digital signatures and certificates, many different operational environments and vast amount of stored data, which require appropriate skills. For this reason corporate administrative and technical directors must check their organization chart to determine who is capable of handling these new processes and services, who is accountable for, and if there is a need to expand staff duties or recruit contractors with specific expertise.

Organizational recommendations often share the same ground that most of the recommendations on human resources have. In fact the implementation of any kind of organizational recommendations can only be ensured if all employees of an organization are familiar with them and aware of the underlying considerations. It will be beneficial to the company to prepare meetings to explain the reason why a different structure was created and who reports to whom.

A further note on organizational recommendations is related to the new proposed regulation on data protection of the European Union (GDPR). The directive requires the definition of new organizational roles, such as the data controllers and data processors, and defines their obligations. Given the already highlighted gaps on privacy (see section 4.2 of this document), a particular attention must be given to these newly created roles and their attribution to a physical person.[243]

Recommendations on human resources (for human resources managing and using virtual environment assets)

Human resources are always considered the main source of threats. They include internal and external users that attack systems either maliciously or accidentally. To limit these issues, as mentioned earlier, all involved parties must focus on training and education of their staff, and put in place assurance processes for the evaluation of human resources.

Big players, such as vendors of virtualization suites, should support specific education initiatives on virtual environments to raise and train tomorrow's engineers, and foster information and communication technology security awareness and training programs. Some certification programs are already available as

[242] http://www.cloud-council.org/deliverables/CSCC-Cloud-Security-Standards-What-to-Expect-and-What-to-Negotiate.pdf, p. 25

[243] For example the obligations of a data processor include: to maintain a written record of processing activities carried out on behalf of each controller; to designate a data protection officer where required, to appoint a representative (when not established in the EU) in certain circumstances, and to notify the controller on becoming aware of a personal data breach.

vendors and industry bodies provide them.[244] It is much more difficult to find vendor-agnostic training courses,[245] and as a result, we urge that such training be through fully-funded independent statutory bodies and at arm's length from vendors of all kinds.

Small and medium private companies, corporations and governmental bodies should encourage their technical staff to attend courses from respected institutes to increase competences. Administrators and other privileged users should cooperate with the international community to exchange information on threats and promote the application of mitigation measures, such as the good practices presented in Section 3.

Moreover, administrators should report on their implementation choices of good practices in terms of considered components, threat, countermeasures, and identified gaps. Final users should learn about their rights and threats to privacy in virtualized environments attending specific courses and educational initiatives.

To conclude, education programs for raising awareness on virtualization security need to be introduced for different types of people, ranging from developers, to administrators and policy makers, to simple users. Developers should be provided with practical courses on the management and implementation of security-aware systems, tools and components. Policy makers and regulators should be aware of security issues, to drive other users in the correct management of virtualization security. Service providers need to understand implications of security considerations when selecting products and services. Schools and universities should prepare well in advance students in all domains (not only in the ICT domain) to be ready to manage security requirements, needs, and technologies.

[244] See for example VMware education and certification program for its virtualization product suite (in http://mylearn.vmware.com/mgrreg/index.cfm) and the Amazon Web Services (AWS) Certifications for Xen products in cloud (in https://aws.amazon.com/certification/certification-prep/).
[245] Some training is proposed in the more generic cloud scenario.

Annex A: Table of weakness groups

The following table presents the selected weakness groups together with the relative CWEs (hierarchically relevant CWE in bold) and a detailed description.

WEAKNESS GROUP	RELATED CWE	DESCRIPTION
Injection	**Injection (CWE-74)** Code injection (CWE-94), OS command injection (CWE-78) SQL command injection (CWE-89) CRLF injection (CWE-93) Externally-controlled format string (CWE-134)	This group refers to weaknesses based on the lack of verification of assumptions on user-controlled input, allowing alteration of the execution by sending code (injection) through legitimate data channel. The injected data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
Improper Authentication	**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) Improper restriction of the communication channel between the endpoints (CWE-923)	This group refers to weaknesses related to absence of restriction or incorrect restriction access to a resource from an unauthorized actor. It affects both specifications and enforcement and includes weakness related to authentication, authorization, user management and restrictions on communication channel between end-points.
Management of credentials	**Credentials Management (CWE-255)** Weak cryptography (CWE-261), Weak password recovery mechanism for forgotten password (CWE-640) Insufficiently protected credentials (CWE-522) Hard-coded credentials (CWE-798)	This group refers to weaknesses related to the management of credentials. It includes weaknesses related to password managements like weak cryptography, aging but also weak password recovery mechanism for forgotten password. It also refers to insufficiently protected credentials both at rest and in transit (i.e., plaintext storage or unprotected transport).
Permissions and privileges management	**Permissions, Privileges and Access Control (CWE-264)** Privilege and sandbox Issues (CWE-265) Permission issues (CWE-275)	This group refers to weaknesses related to the management of permissions, privileges, and other security features that are used to perform access control. More specifically it includes issues related to executions with unnecessary privileges or incorrect privilege assignment dropping/lowering errors and insecure/preserved inherited permissions.
Cryptographic Issues	**Cryptographic Issues (CWE-310)** Key management errors (CWE-320)	This group refers to weaknesses related to the use of cryptography, and in particular to cryptographic errors due to poor design or

WEAKNESS GROUP	RELATED CWE	DESCRIPTION
	Missing encryption of sensitive data (CWE-311) Missing required cryptographic step (CWE-325).	implementation including plaintext storage/transmission of sensitive information, key management errors like key exchange without entity authentication or expired key. It also refers to missing encryption of sensitive data including clear text storage and transmission.
Data handling	**Data handling (CWE-19)** Representation errors (CWE-137) Numeric errors (CWE-189)	This group refers to weaknesses related to functionalities that process data. It is a broad category including, string and type errors, generic representation errors like improper handling of syntactically invalid structure and numeric errors (e.g. wrap-around error, incorrect conversion between numeric types etc.).
Information management errors	**Information management errors (CWE-199)** Information Exposure (CWE-200)	This group refers to weaknesses related to improper handling of sensitive information and in particular information exposure which is the intentional or unintentional disclosure of information to an actor that is not explicitly authorized to have access to that information a.k.a. information leak.
Improper Input Validation	**Improper Input Validation (CWE-20)** Path traversal (CWE-22) Link Following (CWE-59) Memory buffer (CWE-119)	This group refers to improper input validation meaning that the system does not validate or incorrectly validates input. More specifically it refers to pathname traversal and equivalence errors including improper link resolution before file access ('Link Following'). It also includes memory buffer weakness like classic buffer overflow and out-of-bound read or write issues to name but a few.
Insufficient Verification of Data Authenticity	**Insufficient Verification of Data Authenticity (CWE-345)** Cross-Site Request Forgery (CWE-352) Improper verification of cryptographic signature (CWE-347)	This group refers to not sufficiently verified origin or authenticity of data causing acceptance of invalid data. It includes improper verification of cryptographic signature, missing or improper validation of integrity check and Cross-Site Request Forgery (CSRF). CSFR implies that the application does not, or cannot, sufficiently verify whether a well-formed, valid and consistent request was intentionally provided by the user who submitted the request.
Improper Certificate Validation	**Improper Certificate Validation (CWE-295)** Certificate expiration (CWE-298) Check on revocation (CWE-299) Missing validation (CWE-599)	This group refers to certificate that is not validated, or incorrectly validated allowing eventually man-in-the-middle attack. It includes weaknesses related to improper validation with host mismatch, certificate expiration, revocation or missing validation. It also includes weakness

WEAKNESS GROUP	RELATED CWE	DESCRIPTION
		relative to improper following of certificate's chain of trust.
Use of Insufficiently Random Values	**Use of Insufficiently Random Values (CWE-330)** Insufficient entropy (CWE-331) Predictability problems (CWE-240)	This group refers to weaknesses related to the generation of predictable values in a context requiring unpredictability. It refers to weakness related to insufficient entropy specifically in Pseudo-Random Number Generator (PRNG), predictability problems and the use of cryptographically weak PRNG
Resource Management errors	**Resource Management errors (CWE-399)** Resource locking problems (CWE-411). Channel and path errors (CWE-417) Uncontrolled Resource Consumption (CWE-400)	This group refers to weaknesses related to improper management of system resources eventually bringing to resource exhaustion. It also refers to weakness related to improper resource shutdown or release, double free call leading to modification of unexpected memory locations and many others memory management weakness like improper release of memory before removing last reference a.k.a. 'Memory Leak'.
Cross-site Scripting	**Cross-site Scripting (CWE-79)**	This group refers to user-controllable input that is not neutralized or incorrectly neutralized before it is placed in an output that is used and served to other users. It is specific mainly for web pages.
Race conditions	**Race Condition (CWE-362)**	This group refers to code sequence that can run concurrently with other code, and the code sequence requires temporary, exclusive access to a shared resource, but a timing window exists in which the shared resource can be modified by another code sequence that is operating concurrently.
Environment	**Environment (CWE-2)** Interaction Error (CWE-435)	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 run together.
Configuration	**Configuration (CWE-16)**	This group refers to weaknesses typically introduced during the configuration of the software components.

Annex B: Table of weaknesses vs good practices

This table provides an explicit link between weaknesses identified in section 2 and the corresponding good practices of section 3. A brief description is also provided for each good practice. We note that, in the table, we report only those good practices that are proper of virtualized systems. For conciseness and simplicity, horizontal good practices (e.g., timely install updates) are not reported.

WEAKNESS[246]	DESCRIPTION	GOOD PRACTICES
Injection	Protect input parameters. Utilize appropriate mixture of white-list, black-list, and advanced parsing at all levels of the virtualized environment	PL-03 Information classification PL-05 Use segregation in networks G-01: Secure all elements of a full virtualization solution and maintain their security. G-05: Isolation of guest OS and partitioning. G-07: Properly manage images and snapshots. G-08: Vulnerability analysis. HY-13: A functional hypervisor management console with disk footprint and smaller number of exposed interfaces is easy to be verified MISC-02: Secure of offline and dormant VMs MISC-04: Monitor the risk due to cloud service provider API
Improper Access Control	Compartmentalize the system. Do not allow sensitive data to be accessed and/or released to the outside. Reinforce privilege separation.	PL-04 Business requirements of access control, user access management, and system and application access control to avoid abuse of authorizations. PL-05 Use segregation in networks. G-02: Restrict and protect administrator access to the virtualization solution G-05: Isolation of guest OS and partitioning. G-11: Controlled access to VMs C-03: Authorisation and proper documentation of change. HY-02: Restrict administrative access to the management interfaces of the hypervisor HY-20: Mechanism for security monitoring and security policy enforcement of VM operations HY-22: The VM administration access control solution should have granular capability HY-25: The remote access protocol used to access the hypervisor console should have configuration options to deny access

[246] Weaknesses that are considered very important for visualized environments are reported in bold.

WEAKNESS[246]	DESCRIPTION	GOOD PRACTICES
		OS-03: Follow the recommended practices for managing the physical OS, e.g., time synchronization, log management, authentication, remote access, etc. OS-07: Use separate authentication solutions for each guest OS
Management of credentials	Protect and manage credentials in a secure and trusted way	PL-01: use of cryptography PL-02 User awareness through education and training. PL-03 Information classification G-02: Restrict and protect administrator access to the virtualization solution G-07: Properly manage images and snapshots. G-09: Implementation of network best practices. G-11: Control access to VMs C-04: Use configuration audit and control HY-08: Carefully monitor the hypervisor itself for signs of compromise HY-23: The number of user and privileged accounts requiring direct access to hypervisor host should be limited to bare minimum HY-30: Use a dedicated virtual network segment to protect VM management and hypervisor, OS-01: protect sensitive data OS-03: Follow the recommended practices for managing the physical OS, e.g., time synchronization, log management, authentication, remote access, etc. OS-09: If a guest OS is compromised, Assume that all guest OSs on the same hardware VN-04: Protect operational reference data MISC-03: Workload of different trust levels Located on the same server
Permissions and privileges management	Carefully handling permissions and privileges including different administrative levels	PL-03 Information classification G-02: Restrict and protect administrator access to the virtualization solution G-03: Ensure that the hypervisor is properly secured C-03: Document change of authorisation C-04: Configuration audit and control OS-01: protect sensitive data OS-03: Follow the recommended practices for managing the physical OS, e.g., time synchronization, log management,

WEAKNESS[246]	DESCRIPTION	GOOD PRACTICES
		authentication, remote access, etc.HY-02: Restrict administrative access to the management interfaces of the hypervisor HY-23: The number of user and privileged accounts requiring direct access to hypervisor host should be limited to bare minimum HY-24: The user and privileged accounts on the hypervisor must be integrated with the enterprise directory infrastructure
Cryptographic Issues	Handling issues with cryptographic mechanisms ensuring correctness and strangeness.	G-9: Implement network best practices PL-01 The use of cryptography PL-03 Information classification
Data handling	Ensure proper handling of data, by specifying proper management policies and by monitoring misbehaviours. Plan business processes before implementing them.	G-13: Organisational policy for VM security. C-06: Event monitoring OS-09: If a guest OS is compromised, Assume that all guest OSs on the same hardware HY-06: Consider using introspection capabilities to monitor the security of each guest OS HY-07: Consider using introspection capabilities to monitor the security of activity occurring between guest OSs HY-08: Carefully monitor the hypervisor itself for signs of compromise HY-19: The VM image library should reside outside of the hypervisor host, the library should have strict access control HY-29: Generate, if possible, logs in a standardized format to help leverage the use of tools with good analytical capabilities
Information management errors	Careful and secure management of sensitive data. Any information that is not necessary to the working of the system should be removed. Compartmentalize the system.	PL-03 Information classification G-01: Secure all elements of a full virtualization solution and maintain their security G-04: Carefully plan the security for a full virtualization solution before installing, configuring, and deploying it G-05: Isolation of guest OS and partitioning. OS-01*:* protect sensitive data OS-02: secure of pre-configured / active VMs. OS-07: Use separate authentication solutions for each guest OS OS-09: If a guest OS is compromised, Assume that all guest OSs on the same hardwareHY-24: The user and privileged accounts on the hypervisor must be integrated with the enterprise directory infrastructure
Improper Input Validation	Understand all the potential interfaces where untrusted inputs	G-08: Vulnerability analysis. OS-01: protect sensitive data

WEAKNESS[246]	DESCRIPTION	GOOD PRACTICES
	enter the system. Assume all inputs are malicious.	
Insufficient Verification of Data Authenticity	Improve the data authenticity verification	G-02: Restrict and protect administrator access to the virtualization solution. G-13: Organisational policy for VM security.
Improper Certificate Validation	Certificates should be carefully managed and checked	HY-03: Synchronize the virtualized infrastructure to a trusted authoritative time server HY-14: The hypervisor should have a boot configuration choice to disallow the user of non-certified drivers MISC-03: Workload of different trust levels Located on the same server
Use of Insufficiently Random Values	Carefully monitor the entropy provided by the virtualized system when used for encryption or other security features	PL-02 User awareness through education and training C-06: Event monitoring HY-06: Consider using introspection capabilities to monitor the security of each guest OS HY-07: Consider using introspection capabilities to monitor the security of activity occurring between guest Oss
Resource Management Errors	Recognize resource exhaustion problems. Configuration comparison and checks. Quoting and resource allocation planning.	PL-05 Use segregation in networks. G-06: Monitoring of the resources. G-9: Implement network best practices C-02: Hypervisor configuration checks. C-06: Event monitoring HY-04: Disconnect unused physical hardware HY-15: The ratio of the combined configured memory of all VMs to the RAM HY-16: The hypervisor should guarantee physical RAM HY-17: The number of virtual CPUs allocated to any VM deployed should be strictly less than the total number of cores in the hypervisor host HY-18: The hypervisor should provide features to specify lower and upper bound for CPU clock cycles MISC-01: Control the proliferation of VMs MISC-02: Secure offline and dormant VMs
Cross-site Scripting	Understand the context or data usage, and all potential interfaces where untrusted inputs enter the system. Adopt standard	PL-02 User awareness through education and training PL-05 Use segregation in networks. G-08: Vulnerability analysis

WEAKNESS[246]	DESCRIPTION	GOOD PRACTICES
	countermeasure like sessions, packet filtering, firewalling solutions.	OS-09: If a guest OS is compromised, Assume that all guest OSs on the same hardware HY-09: Improve visibility and controls over virtual networks
Environment & Configuration	Carefully evaluate configuration assurance and keep track on the configuration changes, even authorized changes. Consider to work always in untrusted environments.	G-01: Secure all elements of a full virtualization solution and maintain their security G-03: Ensure that the hypervisor is properly secured G-04: Carefully plan the security for a full virtualization solution before installing, configuring, and deploying it G-10: Prevent single point of failure G-12: Secure the host OS G-13: Organisational policy for VM security G-14: Define and verify SLA's and contract requirements G-15: Security departments should be involved in the definition of SLAs. C-01: Configuration assessment C-02: Hypervisor configuration checks C-03: Document change of authorisation C-04: Configuration audit and control C-05: Approved templates for VM deployments C-06: Event monitoring C-07: Configuration management database (CMDB) OS-02: Secure pre-configured / active VMs. OS-04: Install all updates to the guest OS promptly OS-05: Back up the virtual drives used by the guest OS on a regular basis, using the same policy for backups as is used for non-virtualized computers in the organization OS-06: Disconnect unused virtual hardware in each guest OS OS-08: Ensure that virtual devices for the guest OS are associated only with the appropriate physical devices on the host system, such as the mappings between virtual and physical NICs. OS-10: Investigate each guest OS for compromise, just as one would during normal scanning for malwares CON-01: Secure host. CON-02: Secure containers CON-03: Configure containers properly

WEAKNESS[246]	DESCRIPTION	GOOD PRACTICES
		HY-01: Install all updates to the hypervisor as the vendor releases them HY-03: Synchronize the virtualized infrastructure to a trusted authoritative time server HY-04: Disconnect unused physical hardware HY-05: Disable all hypervisor services such as clipboard- or file-sharing HY-09: Lack of visibility and controls over virtual networks HY-10: A Type I hypervisor provides more security assurance than a Type II hypervisor HY-11: A hypervisor platform with hardware-assisted virtualization provides greater security assurance than hypervisors with purely software- assisted virtualization. HY-12: The hypervisor should be part of an overall infrastructure HY-13: A functional hypervisor management console with disk footprint and smaller number of exposed interfaces is easy to be verified HY-14: The hypervisor should have a boot configuration choice to disallow the user of non-certified drivers HY-19: The VM image library should reside outside of the hypervisor host HY-21: Solutions for the security monitoring and the security policy enforcement of the production VMs should be based "outside of VMs" HY-26: Always use hypervisor features that enable the definition of a "gold configuration" HY-27: A hypervisor patch management practice must be in place. HY-28: Configure the built-in hypervisor firewall HY-29: Generate, if possible, logs in a standardized format HY-31: Communications from a given VM to the physical network should be enabled by establishing multiple communication paths within the virtualized host. VN-01: Clearly define security dependencies and trust boundaries VN-02: Assure robust identity. VN-03: Build security on open standards VN-04: Protect operational reference data VN-05: Make systems secure by default VN-06: Provide accountability and traceability

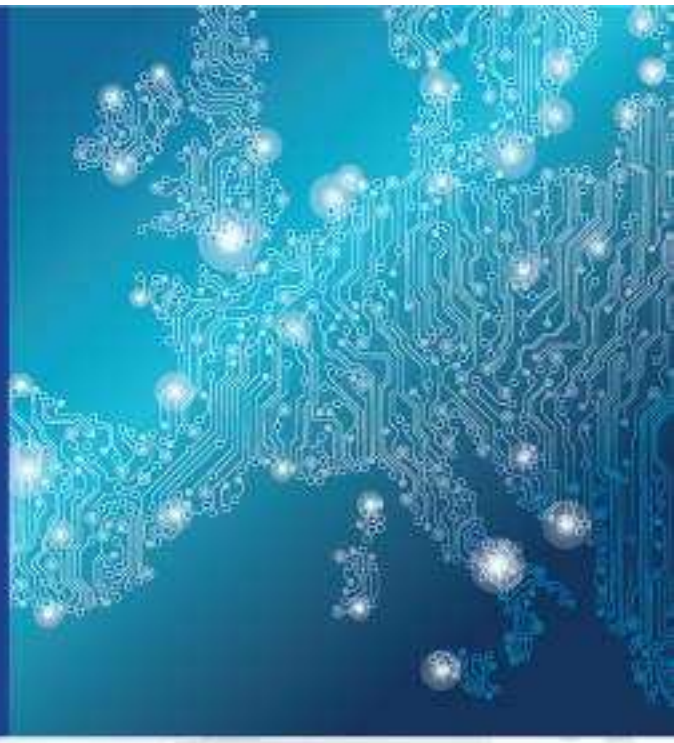
WEAKNESS[246]	DESCRIPTION	GOOD PRACTICES
		VN-07: Properties of manageable security controls

ENISA

European Union Agency for Network
and Information Security
Science and Technology Park of Crete (ITE)
Vassilika Vouton, 700 13, Heraklion, Greece

Athens Office

1 Vass. Sofias
Marousi 151 24, Athens, Greece



TP-05-16-091-EN-N

PO Box 1309, 710 01 Heraklion, Greece
Tel: +30 28 14 40 9710
info@enisa.europa.eu
www.enisa.europa.eu

ISBN: 978-92-9204-211-0
DOI: 10.2824/955316