An empirical analysis of vulnerabilities in virtualization technologies

Antonios Gkortzis*†, Stamatia Rizou†, and Diomidis Spinellis*

* *Department of Management Science and Technology*
Athens University of Economics and Business
Athens, Greece
{*antoniosgkortzis, dds*}*@aueb.gr*

† *European Projects Department*
Singular Logic S.A.
Athens, Greece
srizou@singularlogic.eu

***Abstract*—Cloud computing relies on virtualization technologies to provide computer resource elasticity and scalability. Despite its benefits, virtualization technologies come with serious concerns in terms of security. Although existing work focuses on specific vulnerabilities and attack models related to virtualization, a systematic analysis of known vulnerabilities for different virtualization models, including hypervisor-based and container-based solutions is not present in the literature. In this paper, we present an overview of the existing known vulnerabilities for hypervisor and container solutions reported in the CVE database and classified under CWE categories. Given the vulnerability identification and categorization, we analyze our results with respect to different virtualization models and license schemes (open source/commercial). Our findings show among others that hypervisors and containers share common weaknesses with most of their vulnerabilities reported in the category of security features.**

1. Introduction

Cloud computing is currently considered a well-established solution to the increasing demand of computing resources, since it enables elasticity of IT services and facilitates their administration at large scale. Virtualization is the main software technology used to support the cloud computing model. Its goal is to manipulate a shared pool of configurable computing resources and allow the coexistence of several isolated systems on a single physical machine [**?**]. However, virtualization comes with a number of considerations with respect to security. Compared to the bare metal servers, the use of Virtual Machines (VMs) increases the attack surface of the system [**?**] since virtualization technologies provide access for multiple services to the same physical resources. A vulnerability exploited in virtualized multi-tenant systems can cause data-leakage, denial-of-service or violation of privacy [**?**] issues to more than one entities. Thus, the need for securing these software components is great and the process for achieving this goal is far from trivial.

A first step towards securing the virtualized environments is the better understanding of the underlying virtualization technologies and their respective vulnerabilities. Common technologies for virtualization include the use of hypervisors and containers. In general, virtualized environments use typically a hypervisor, which constitutes a piece of software mediating between bare metal and VMs to ensure the proper execution and management of VMs on the host machine. Recently, containers which refer to another type of virtualization at the OS level, have become popular in cloud computing environments. Containers offer a variety of benefits compared to the hypervisors, especially in terms of performance and scalability, but they are considered to be more vulnerable in terms of security [**?**]. Although there have been several studies that compare the performance between hypervisors and containers [**?**], there is little evidence in the literature on the security characteristics of both hypervisor and container technologies.

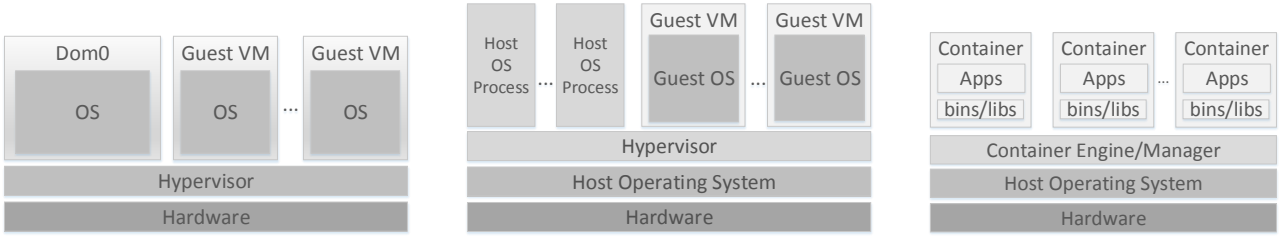
This paper aims to contribute towards this direction by providing a comprehensive overview of the existing identified vulnerabilities for virtualization technologies. To achieve our goal, we have conducted an extensive study of the known reported vulnerabilities for each of the most popular hypervisor and container solutions. In our study, we have reviewed the vulnerabilities that we collected via the Common Vulnerabilities and Exposures (CVE) database (http://cve.mitre.org/) for eight popular open source and commercial hypervisors and five emerging container management solutions. For the classification of the known vulnerabilities, we have used the categories of the Common Weakness Enumeration (CWE) database (http://cwe.mitre.org/) to map each vulnerability with a concrete description. Our results show that most of the known vulnerabilities for hypervisors fall under the categories of *Data Handling* and *Security Features*. Furthermore, we observe a similar trend with respect to the vulnerabilities concerning security and in particular the implementation of *Permissions, Privileges and Access Controls* in the case of container-based virtualization.

The structure of the rest of the paper is as follows: In Section 2, we present the different virtualization technologies and their implementations. Next, in Section 3 we present the methodology for identifying and classifying the vulnerabilities. In Section 4 we present the results and we discuss our findings. In Section 5 we present the related work and finally, in Section 6 we conclude the paper.

2. Virtualization Technologies

In this Section we present the different virtualization technologies, the hypervisor-based and the container-based,

978-1-5090-1445-3/16/$31.00 2016 IEEE (CloudSPD'16)



(a) Hypervisor-based type I virtualization (b) Hypervisor-based type II virtualization (c) Container-based virtualization

Figure 1: Virtualization technologies' architectures

that exist and can be used to enable the cloud model. Furthermore, we explain the scope of our study with respect to the selected hypervisor and container solutions.

2.1. Virtualization Models

Hypervisors are software components that allow the deployment, configuration and management of several VMs on a system. Depending on whether they need a host operating system to be present or not, they can be classified to type-I and type-II hypervisors. On the one hand, type-I hypervisors or *bare metal* are deployed directly on the machine and manages its recourses, as shown in Figure 1a. Most type-I hypervisors require a *Dom0* (domain zero) which is a privileged guest system that acts as a manager for the other unprivileged guest systems. On the other hand, type-II hypervisors or *hosted* require a host operating system to be present. This type of hypervisor is installed and operates on top of the host operating system as any other user-space application as shown in Figure 1b. Type-II hypervisors are very easy to use and can provide, to some extent, the same functionalities as type-I hypervisors do. The downside of this virtualization technology is the overhead that they introduce due to the need to consume resources in order to virtualize hardware. We would like to note that despite the clear definition of type-I and type-II hypervisors, in practice, the distinction between the two hypervisor types is not always that clear. For example, hypervisors, such as Kernel-based Virtual Machine (KVM) [1] and FreeBSD bhyve[2] can be installed and operate as a Kernel module making the hosts operating system behave as type-I hypervisor, while the need for a host operating system makes them valid candidates for type-II hypervisor class as well.

Containers is another type of virtualization that has recently been adopted by the cloud community. As shown in Figure 1c, containers are co-existing user-space instances with a group of isolated processes, that share the kernel of an operating system. Since they do not need to virtualize hardware, containers produce little to no overhead and offer more dense deployment than virtual machines [**?**]. In contrary to the hypervisors, containers cannot host guest operating systems with different Kernel and thus, provided services can be limited to this Kernel's capabilities.

2.2. Scope of the Study

For each virtualization model presented in the previous paragraph we listed several open-source and proprietary software solutions, and we then laid down the scope of our study by selecting those solutions that are more popular and for which vulnerabilities are known. In this line, we omitted software solutions that were outdated (BSD Jails [3]), discontinued (VMware Server [4] and Let Me Contain That For You (lmctfy) [5]) or bound to very specific hardware (IBM z/VM [6] and Oracle VM Server for SPARC [7]). Table 1 summarizes the main characteristics of the hypervisors selected for our study. As it is shown, the selected set of hypervisors provide a good mixture of open source and commercial solutions

TABLE 1: Overview of the selected Hypervisor solutions

Hypervisor	License	Type	Year of Release
VMware Workstation [8]	Proprietary	Type II	1999
VMware ESXi [9]	Proprietary	Type I	2002
Xen [10]	Open Source	Type I	2003
Oracle VirtualBox [11]	Open Source	Type II	2007
Microsoft Hyper-V [12]	Proprietary	Type I	2008
XenServer [13]	Proprietary	Type I	2008
Red Hat Enterprise Virtualization (RHEV) [14]	Proprietary	Type I	2010
KVM	Open Source	Type I/ Type II	2012

1. http://www.linux-kvm.org
2. http://bhyve.org/
3. https://www.freebsd.org/cgi/man.cgi?query=jail
4. https://my.vmware.com/web/vmware/info?slug=infrastructure_operations_management/vmware_server/2_0
5. http://lmctfy.io/
6. http://www-03.ibm.com/systems/power/software/virtualization/
7. http://www.oracle.com/us/technologies/virtualization/oracle-vm-server-for-sparc/overview/index.html
8. http://www.vmware.com/products/workstation.html
9. https://www.vmware.com/products/esxi-and-esx.html
10. http://www.xenproject.org/
11. https://www.virtualbox.org/
12. https://www.microsoft.com/en-us/cloud-platform/virtualization
13. http://xenserver.org/
14. https://www.redhat.com/en/technologies/virtualization

TABLE 2: Vulnerabilities' categories

CWE Id	Vulnerability category	Description
CWE-16	**Configuration**	Weaknesses referring to software (e.g. Server & Application) misconfiguration. *Configuration* is a super-category at the same level (2) as the following *Code* category.
CWE-17	**Code**	Weaknesses that are introduced in a system during the inception, design and implementation life-cycle phases. *Code* is a level-2 super-category of the following twenty-one categories.
CWE-19	**Data Handling**	Weaknesses related to functionalities that processes data. *Data Handling* is a level-4 super-category of the next eight listed categories.
CWE-200	*Information Exposure*	Weaknesses related to information disclosure to unauthorized actors. Information leak might be both intentional and unintentional.
CWE-20	*Improper Input Validation*	Lack or incorrect validation of input. Unexpected input can affect the program's flow or even execute arbitrary code.
CWE-94	*Code Injection*	Lack or incorrect neutralization of special elements of an input that might cause unexpected results when executed.
CWE-79	*Cross-site Scripting*	Lack or incorrect neutralization of user-controllable input before it is placed as an output web page served to other users.
CWE-119	*Impr. Restr. of Operations within the Bounds of a Memory Buffer*	Accessing memory locations outside of the program-intended boundaries, allowing the execution of arbitrary code, alteration of the program's flow, system crashes or leakage of sensitive information.
CWE-22	*Path Traversal*	Weaknesses that fail to neutralize special elements within a pathname. This can cause the pathname to resolve to a location that is outside of the restricted directory.
CWE-189	*Numeric Errors*	Weaknesses related to improper calculation or conversion of numbers.
CWE-59	*Link Following*	Weaknesses that allow unauthorized access to unintended recourses when resolving file links or shortcuts.
CWE-398	**Indicator of Poor Code Quality**	Features that indicate a software product has not been carefully developed or maintained. *Indicator of Poor Code Quality* is a level-4 super-category of the *Resource Management Errors* category.
CWE-399	*Resource Management Errors*	Weaknesses related to improper management of system resources.
CWE-254	**Security Features**	Weaknesses related to topics like authentication, access control, confidentiality, cryptography, and privilege management. *Security Features* is a level-4 super-category of the next seven listed categories.
CWE-264	*Permissions, Privileges, and Access Controls*	Weaknesses related to the management of permissions, privileges, and other security features that are used to perform access control.
CWE-255	*Credentials Management*	Weaknesses related to the management of credentials.
CWE-310	*Cryptographic Issues*	Weaknesses related to the use of cryptography.
CWE-284	*Improper Access Control*	Fail to restrict access to a resource to an unauthorized actor.
CWE-287	*Improper Authentication*	Fail to authenticate an actor's identity.
CWE-352	*Cross-Site Request Forgery (CSRF)*	The web application does not sufficiently verify whether a well-formed, valid, consistent request was intentionally provided by the user who submitted the request.
CWE-89	*SQL Injection*	Weaknesses related to incorrect detection of SQL commands built by external parameters for malicious causes.
CWE-361	**Time and State**	Weaknesses in this category are related to the improper management of time and state in an environment that supports simultaneous or near-simultaneous computation by multiple systems, processes, or threads. *Time and State* is a level-4 super-category of the *Race Condition* category.
CWE-362	*Race Condition*	Condition under which a process acquires 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.

delivering different types of virtualization models. It is also worth to note that hypervisor solutions like Windows Virtual PC [15] and FreeBSD bhyve were analyzed but they were not finally included in the study results since we did detect no relevant known vulnerabilities.

TABLE 3: Overview of the selected Container solutions

Container	License	Year of Release
Open Virtuozzo (OpenVZ) [16]	Open Source	2005
Linux Containers (LXC) [17]	Open Source	2008
Docker [18]	Open Source	2013
Kubernetes [19]	Open Source	2014
LXD [20]	Open Source	2015

In the case of containers, the set of available solutions has been more limited due to the recent adoption of this technology from the cloud community, which basically started from 2013. Beyond Docker, which is currently the dominant container implementation, we have also included in our study some other less popular solutions like LXC, LXD to provide a broader scope for our study. The selected container solutions are summarized in Table 3. As in the case of hypervisors, there were also some container-based virtualization solutions like CoreOS [21], Warden Container [22], and Proxmox Virtual Environment (VE) [23] (with the latter representing a special case since it can deploy and manage

15. https://www.microsoft.com/en-us/download/details.aspx?id=3702

16. https://openvz.org
17. https://linuxcontainers.org/
18. https://www.docker.com/
19. http://kubernetes.io/
20. https://linuxcontainers.org/
21. https://coreos.com/blog/rocket/
22. https://github.com/cloudfoundry/warden
23. https://www.proxmox.com/en/

both virtual machines as type-I hypervisor and containers) that although they were analyzed, no vulnerabilities were finally detected.

3. Study Methodology

In this Section, we present the methodology followed in our study for retrieving and analyzing the known vulnerabilities. Data from three different databases were retrieved and combined in order to detect and classify the known vulnerabilities for the different virtualization technologies. In particular, we used the CVE database to identify the known vulnerabilities, the CWE database to retrieve the categories of the weaknesses and the CVE-details database (http://www.cvedetails.com/) to classify the vulnerabilities in different categories. The first step has been to parse the CVE database which constitutes a dictionary of common names (i.e., CVE Identifiers) for publicly known cybersecurity vulnerabilities and is widely accepted by the major software industry players. The CVE entries were filtered based on keywords related to the virtualization technologies and solutions presented in Section 2 such as Xen, KVM, Hyper-V, Docker, etc. This step resulted to the identification of 983 possible relevant vulnerabilities that were later on manually parsed to detect the false positive matched CVEs, leading thus to rejecting more than one third of the dataset.

Having at hand the filtered and manually confirmed CVE entries, we then parsed the CWE database to retrieve the weaknesses categories and match the known vulnerabilities in different CWE categories according to the CVE-details information. CWE is an hierarchical tree-based schema of different vulnerability categories, which consists in total of 244 categories and 719 subcategories. To identify the corresponding category for each detected CVE entry, we retrieved the CWE id for the selected CVEs from the CVE-details dataset. In general, we have considered the following rationale for the categorization of the CVE entries in the CWE categories: 1) We categorized each CVE entry in the most detailed CWE. Thus, the CVE entries that were lacking specific details have been categorized into higher level categories. 2) When analyzing the results vulnerabilities were counted in their actual reported by CVE-details category and not summed under their super-categories (highlighted in Table 2). This process resulted to the assignment of the majority of the detected vulnerabilities in twenty three distinct CWE categories. To keep the paper self-contained, we present in Table 2, brief explanations of the twenty one categories based on their CWE definitions.

4. Study Results & Discussion

In this Section we present the results of the analysis performed on the dataset created with the process described in Section 3. The distinct vulnerabilities that matched our hypervisor and container related keywords were 524. Adding those that affected more than one virtualization technology the results sum up to 540 for the hypervisor related keywords and 34 for the container related keywords.

Figure 2 shows the distribution of the vulnerabilities with respect to the year reported for both hypervisor and container solutions. Overall, we observe an increasing trend in the number of vulnerabilities for the virtualization technologies, which is expected given the wide use of virtualization over the last years. It is important to note here that we have studied vulnerabilities reported until July 2016 which can explain, to some extent, the drop in the numbers of 2016 in Figure 2. Furthermore, the number of reported vulnerabilities for containers is significantly lower than those of the hypervisors, which can be justified from the fact that containers is still an emerging solution for lightweight virtualization.

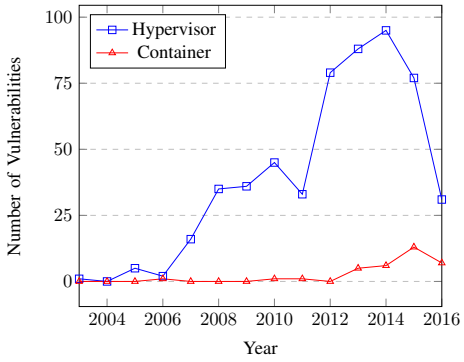


Figure 2: Vulnerabilities' per year variation

Table 4 provides a comprehensive breakdown of the detected vulnerabilities for the different hypervisors presented in Section 2. Note that in Table 4, each vulnerability is counted in the most specific CWE category and then summed to their super-category (presented with bold font) including vulnerabilities that were directly assigned to these super-categories. Super-categories, like *Indicator of Poor Code Quality* and *Time and State* have no vulnerabilities assigned, since CVE entries were assigned only to their subcategories.

In terms of the vulnerability types, we observe that the majority of the total reported vulnerabilities for the hypervisors belong in the following four categories: 1) *Permissions, Privileges and Access Controls* is the dominant category attracting almost 20% (105 out of 540) of the total vulnerabilities; 2) *Improper Input Validation* follows with 13% (73 out of 540) of the total vulnerabilities; 3) *Improper Restriction of Operations within the Bounds of a Memory Buffer* have approximately 10% (53 out of 540) of the reported vulnerabilities; 4) *Resource Management Errors* finally, attracts approximately 10% (56 out of 540) of the total vulnerabilities. *Data Handling* and *Security Features* are the most prone to vulnerabilities super-categories. This global trend in the four most popular categories and the two aforementioned super-categories is evident also in the results for most of the hypervisors, except for VirtualBox, for which we lack details on the characterization of the vulnerabilities that makes it difficult to interpret its results. Hyper-V has also a small number of vulnerabilities reported compared to other hypervisors. However, it can be said that it follows the global trend observed with most vulnerabilities

TABLE 4: Vulnerabilities of Hypervisors

Vulnerability category	VMware ESX	KVM	Xen	Xen Server	HyperV	RHEV	VM Workstation	Virtual Box	Total
Configuration	**1**	**1**	**4**	**1**	-	-	-	-	**7**
Code	-	-	**9**	-	**1**	**1**	-	-	**11**
Data Handling	**30**	**30**	**93**	**17**	**6**	**8**	**8**	**1**	**195**
Information Exposure	3	2	15	-	2	3	1	-	26
Impr. Input Validation	8	12	41	2	3	2	4	1	71
Code Injection	1	-	-	3	-	-	1	-	5
Cross-site Scripting	4	-	1	4	-	1	-	-	10
Impr. Restr. of Operations within the Bounds of a Memory Buffer	10	11	23	9	1	2	2	-	58
Path Traversal	3	-	-	-	-	-	-	-	3
Link Following	-	-	2	-	-	-	-	2	4
Numeric Errors	1	5	10	-	-	-	-	-	16
Indicator of Poor Code Quality	**4**	**11**	**33**	-	**1**	**1**	**1**	**2**	**53**
Resource Management Errors	4	11	33	-	1	1	1	2	53
Security Features	**17**	**18**	**54**	**18**	**3**	**17**	**9**	**3**	**139**
Perms, Privileges, and Access Ctrls	11	15	47	8	1	13	8	2	105
Credentials Management	-	1	-	-	-	2	-	-	3
Cryptographic Issues	2	1	-	1	-	2	-	-	6
Improper Access Control	1	-	4	4	1	-	1	-	11
Improper Authentication	3	1	-	-	-	-	-	1	5
Cross-Site Request Forgery	-	-	-	3	-	-	-	-	3
SQL Injection	-	-	-	2	-	-	-	-	2
Time and State	**1**	**5**	**1**	-	-	**1**	**1**	-	**9**
Race Condition	1	5	1	-	-	1	1	-	9
"Unknown" (Uncategorized)	23	10	28	8	-	5	10	42	126
Total	76	75	222	44	11	33	29	50	540

TABLE 5: Vulnerabilities of Containers

Vulnerability category	Docker	LXC	LXD	Open-VZ	Kuber-netes	Total
Code	**1**	**1**	-	-	-	**2**
Data Handling	**5**	**3**	-	**1**	**2**	**11**
Numeric Errors		-	-	1	-	1
Information Exposure	-	-	-	-	1	1
Improper Input Validation	2	-	-	-	-	2
Path Traversal	-	-	-	-	1	1
Link Following	3	3	-	-	-	6
Security Features	**7**	**6**	**4**	**2**	**2**	**21**
Perms, Privileges, and Access Controls	5	5	-	1	1	12
Improper Access Control	-	-	-	-	1	1
Cryptographic Issues	-	-	-	1	-	1
"Unknown" (Uncategorized)	2	1	4	-	-	7
Total	13	10	4	3	4	34

reported in the super-categories *Data Handling* and *Security Features*, while it stands out in the vulnerabilities related to *Information Exposure* and *Improper Input Validation*.

With regard to licensing, we observe that open source hypervisors, i.e., Xen, KVM & VirtualBox have significantly more vulnerabilities reported (ranging from 75 up to 222) compared to the reported vulnerabilities for the proprietary code solutions like VM Workstation, XenServer, HREV and VM Workstation. This can be explained by the activity of the open source community and the fact that all actions are public. Also, security-related assessment in proprietary solutions is usually performed in-house and some change-logs are used for reporting any bug fixing without giving many details.

Xen, the first widely used open source hypervisor, is leading in the number of reported vulnerabilities with 222 assigned CVEs in total. XenServer, which is a commercial solution, shows less vulnerabilities than Xen due to the fact that it uses Xen's code base and patches are applied in both systems. Notably, several vulnerabilities related to web services, like remote access, which is an extra feature implemented in XenServer have been reported. This can also be observed in the case of RHEV, another commercial hypervisor, for which the reported vulnerabilities in three of the four most popular categories have significantly less reported vulnerabilities. An exception is the *Permissions, Privileges and Access Control* category which attracts almost two times more vulnerabilities than its KVM code-base. This can be explained due to the extra features that RHEV offers as a commercial solution, which is also visible by a manual inspection of the reported vulnerabilities.

Table 5 provides an overview of the vulnerabilities identified for the container-based virtualization solutions. Although the number of known vulnerabilities are significantly lower than those in the case of the hypervisors, some overall trends can be observed (especially for Docker and LXC, which are the most popular solutions). *Permissions, Privileges and Access Controls* is the category with the most reported vulnerabilities with 32.4% (12 out of 34), followed by *Link Following* with 16.21% (6 out of 34). Based also on the hypervisor results, *Permissions, Privileges and Access Controls* is the most popular category in attracting vulnera-

bilities, demonstrating a common global trend affecting both hypervisors and containers.

5. Related Work

Vulnerability study is an initial step for increasing systems security and as such, it has attracted the interest of researchers. There are surveys that study the threats in the Cloud Computing environment [**?**], [**?**] and service delivery models [**?**]. In surveys [**?**] and [**?**] authors study the vulnerabilities reported for the virtualization technology and point out the possible attack surfaces and vectors between the virtualization layers and propose countermeasures.

Vulnerabilities in the Virtual Machine Monitor and Container Management layer, which is the specific focus of our work, has not been extensively studied in the literature. Closer to our work, in study [**?**], authors analyze a dataset of 97 vulnerabilities reports related to the open source Xen and KVM hypervisors. Authors collected their data from four different vulnerability databases using the CVE identifier as a common reference. By studying the source code of the hypervisors and their reported vulnerabilities they mapped the vulnerabilities to 11 hypervisor functionalities and then classified them according to the trigger source and the attack target characteristics. However, this work covers vulnerabilities from only two open source hypervisors and do not consider container technologies. Furthermore, the study reports vulnerabilities up to 2013. In this paper, we have provided a much broader vulnerability study for eight hypervisors and five container technologies providing a more complete up-to-date analysis of vulnerabilities in the virtualization landscape. To the best of our knowledge, this is the first work which specifically attempts a systematic analysis of vulnerabilities for both hypervisors and container technologies.

6. Conclusions

In this paper, we have presented the results of an extensive study which identifies and categorizes known vulnerabilities for different virtualization technologies. Our findings show that in general hypervisor technologies follow a common trend with the majority of the vulnerabilities reported under *Data Handling* and *Security Features* categories. Furthermore, the results show a distinction between open source and commercial hypervisors, with the latter, reporting more vulnerabilities related to extra features, like implementation of web services for XenServer. With respect to the type of hypervisor, in our study we did not observe particular differences in the overall trends of the reported vulnerabilities. Finally, for the container-based virtualization, we observed that the category of *Permissions, Privileges and Access Controls* was assigned the most vulnerabilities as in the case of hypervisors.

Overall, the results of this study can be used from different stakeholders to advance their understanding on the virtualization vulnerabilities. On the one hand, developers and practitioners can use the findings of this study to focus their attention on particular security aspects, which attract the most vulnerabilities such as *Security Features*, *Operations within the Bounds of a Memory Buffer*, *Input Validation*, and *Resource Management and Permissions*, *Privileges and Access Controls*. On the other hand, researchers can use these findings in order to investigate the reason that these categories attract significantly more vulnerabilities than other categories and propose efficient and reliable solutions for detecting and fixing these vulnerabilities. Part of future work in this direction includes the further analysis of the uncategorized vulnerabilities which account for 23% (126 out of 540) of the total vulnerabilities in our study. Furthermore, we believe that our findings could be useful as a baseline for the analysis of different attack models, which ultimately leads to the further securing of virtualization technologies.

Acknowledgments

The research presented in this paper is supported by the European Union within the H2020 MSCA-ITN-EID Project "SENECA".