Security Assessment for Guest-to-Guest and Host-to-Guest Isolation in Type 1 and Type 2 Open-Source Hypervisors: A Focus on Attack Vectors

Said Ally
The Open University of Tanzania, Tanzania
said.ally@out.ac.tz

Abstract
The global IT management landscape has shifted from physical to virtual computing. The transition process that makes virtualized execution environments possible is controlled by the underlying software apparatus known as a hypervisor. Based on the design architecture and configuration, hypervisors differ in the degree of virtual machine isolation, making security a serious concern for technology adopters. This paper presents the security assessment of open-source hypervisors using attack vectors for guest-to-guest (G2G) and host-to-guest (H2G) penetrations. The study uses Proxmox VE and XenServer for Type 1 hypervisors and Kernel Virtual Machine (KVM) and Oracle Virtual Box (OVB) for Type 2, with secondary data analysis based on software vulnerabilities and exposures retrieved from publicly available online databases. For clarity, the source codes of each hypervisor were scanned to identify vulnerable files in an experiment conducted on a Kali Linux testbed with prebuilt virtual machines, each hosting one hypervisor. The vulnerability level was determined using 11 attack vectors extracted qualitatively from relevant literature. The soft memory management unit was found to be the most common attack vector among all hypervisors. Type 1 hypervisors are far better at responding to virtual resource attacks, whereas type 2 hypervisors are more vulnerable to attacks that suffocate computational resources, especially virtual CPUs. OVB outperforms other hypervisors in terms of disk and network performance as it is more resistant to attacks involving I/O networking, interrupt and timer mechanisms, and hypercalls. The results also show that all hypervisors perform better against G2G than H2G attacks. For H2G attacks, the Proxmox VE and KVM have demonstrated better performance compared to other hypervisors. According to analysis, the most prevalent hypervisor flaws are mainly due to design faults rather than misconfigurations by adopters. To get rid of hypervisor weaknesses and fully capitalize on the technological shift from physical to virtual

computing, adopters should consider industry-accepted best practices when selecting, installing, and deploying open-source hypervisors.
Keywords: *Virtual Machine Isolation, Hypervisor Security, Open-Source Virtualization, Attack Vector*

INTRODUCTION

As more businesses become fully dependent on cloud services, IT infrastructure shifts from physical to virtual computing to maximize utilization of hardware resources while drastically reducing operational costs on space, labor, power, and cooling expenses. Virtualization is also popular because of its advantages in mobility, portability, and easy software management (Stoyanov, 2024). The value of open-source software (OSS) is growing dramatically (Hoffmann et al., 2024), with reports showing that it currently leads in server applications. Using OSS solutions in the virtualization process provides further economic benefits (Liang et al., 2024; Gentile et al., 2024; Duy et al., 2024), taking advantage of their General Public License (GPL-GNU), which permits free software access and source code modification (MacDonald, 2013).

Hypervisor software is a main apparatus used to virtualize server machines by creating, controlling, and managing multiple virtual machines running concurrently (Chen et al., 2023). A virtual machine is considered a fully functional computer with a guest operating system and virtual resources for processing (virtual CPU), memory (virtual RAM), storage (virtual HDD), and networking (virtual NIC) capabilities. Virtual machines are created to run on a single host system, with static, dynamic, and on-demand reallocation of server resources (Sinha & Thakare, 2023).

Hypervisors are classified into two types based on their operation mode (Popek & Goldberg, 1974), type 1 and type 2 (Basu et al., 2019; Đorđević et al., 2024). Type 1 runs on hardware (Singh et al., 2016) as embedded, native, or bare metal, while Type 2 is hosted by the operating system (Vojnak et al., 2019), as indicated in Fig. 1.

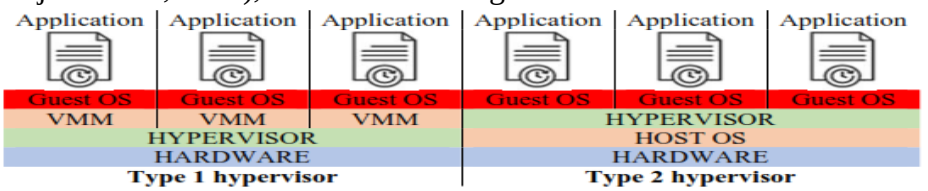

Figure 1*: Hypervisor structure*
Source: Vojtesek & Pipis, 2016; Pandey, 2020

Although type 2 hypervisor is less trusted than type 1 (Obasuyi & Sari, 2015; Felter et al., 2015) as it relies on the operating system, it is evident that the security of virtual machines remains uncertain in both types due to adopters' misconfigurations and hypervisor design faults (Ally et al., 2018). The flaws are largely dependent on the hypervisor-predesigned virtualization method, which includes full virtualization (FV), paravirtualization (PV), hardware-assisted virtualization (HW-aV), and OS-level virtualization (OS-lV) (Rodríguez-Haro et al., 2012; Masood et al., 2014; Zhang et al., 2024).

Both hypervisor types can fully control virtualized safety-critical embedded systems (Lozano et al., 2023), but because they place all files in a single target box to maximize resource utilization and server consolidation, they are considered vulnerable to high-risk attacks (Aalam et al., 2021).

The attack vectors exploit the existing bugs, particularly when the hypervisor source code remains accessible, and the default settings are not well customized to meet security standards. The risk is high in open-source software because of its freely accessible source code (Vainio & Vadén, 2012), its exponential growth, and inherent system bugs and vulnerabilities (Zajdel et al., 2022). Furthermore, when attackers gain full control of the hypervisor and its virtual machines in a cloud environment (Iqbal et al., 2016), which is a typical computing infrastructure, they pose a major security risk (Zoughbi, 2024) as it is complex to spot. Thus, this paper presents a comparative analysis of the security state of type 1 and type 2 open-source hypervisors using attack vectors in a virtual environment. Adopters who wish to reap the benefits of this technological paradigm shift can refer to this study to better grasp the hypervisor design strengths and security configurations from the outset of deployment.

MATERIALS AND METHODS

Extraction of Attack Vectors

To determine the degree of isolation between virtual machines (VMs) and their underlying hosts, the study is designed to use the most common hypervisor attack vectors, extracted qualitatively from relevant literature (Perez-Botero et al., 2013). Using a deductive approach, eleven attack vectors were identified: *soft memory management unit (soft MMU), virtual CPUs, interrupt and timer mechanisms, I/O and networking, hypercalls, remote management software, VM exits, hypervisor add-ons,*

symmetric multiple processors, para-virtualized I/O, and VM management. Each attack vector was mapped to a hypervisor for G2G and H2G penetrations to determine the vulnerability level.

Choice of Hypervisors

The study chose to use economically stable (Freet et al., 2016) open-source hypervisors released under the GNU/GPL license (Tu, 2000). Although different virtualization software provides varying performance (Morabito et al., 2015), the study is set to use the worldwide and most popular open-source solutions (Anwer et al., 2010; Kulkarni et al., 2012; Obasuyi and Sari, 2015; Ally, 2018). For type 1, the selected hypervisors are Proxmox VE (Proxmox VE, 2016, 2018, 2021; Goldman, 2016) and XenServer (XenServer, 2017; 2018). Proxmox VE combines Kernel-based Virtual Machine (KVM) and Virtuozzo (OpenVZ) to support full virtualization and container-based virtualization (Kovari & Dukan, 2012), which are necessary for provision of all major compute, network, and storage functionalities in a single package (Obasuyi & Sari, 2015). XenServer is the most well-known platform used by the world's largest clouds to host mission-critical applications (XenServer, 2017; 2018).

For type 2, kernel-based virtual machines (KVM, 2018, 2021; Hirt, 2010; Kiszka, 2010; Chirammal et al., 2016; Zhang et al., 2021) and Oracle VirtualBox (VirtualBox, 2011, 2018, 2021; Pandey, 2020; Reddy et al., 2022) were used. KVM is Linux-dependent and is the default virtual machine manager for Redhat Enterprise Linux (RHEL). It can run in the CPU's most privileged and protected zone, ring 0. The selected hypervisors include all the features of typical virtualization software. Table 1 compares basic virtualization features for each hypervisor.

Table 1: *Characteristics of the selected hypervisors*

SN	Supported Feature	Type 1		Type 2	
		Proxmox VE	**XenServer**	**KVM**	**OVB**
1	Virtualization Technique	FV, OS-lV	PV, HW-aV	FV, PV, HW-aV	FV, PV, HW-aV
2	Operating System	Linux, MS Windows	Linux, MS Windows	Linux, MS Windows, Unix	Linux, MS Windows
3	Server Architecture	x86, x64	x86, x64	x86, x64	x86, x64
4	Number of VMs	Varies	500	Various	128
5	Number of Virtual CPUs	160	160	Various	Various
6	Maximum RAM per VM	2,000 GB	128 GB	2 TB	1 TB, 16GB
7	Software License (OSS)	Yes	Yes	Yes	Yes
8	Software Version	4.4	4.5.x	2.6.20	5.2
9	Year Released	2016	2016	2007	2015

Vulnerability Analysis

Vulnerability analysis of the selected hypervisors was performed using two techniques: *text search* and *penetration testing tools*. The text search was used to extract the number of hypervisor attack vectors as secondary data from the national vulnerability database (NVD, 2021, 2022), a public repository maintained by the US government. The database uses a system of unique identifiers (Mitre, 2024) known as Common Vulnerabilities and Exposure (CVE).

Only vulnerabilities that matched the search criteria and keywords were returned, including the hypervisor product name, vendor name, CVE assigned unique number, and an open vulnerability and assessment language query. The retrieval process applied an advanced query processing model as a search technique to determine the degree of resemblance and correlation between attack vectors and hypervisor type using a similarity score (Panja, 2024). With an eight-year query period from 2015 to 2023, 11 items exactly matched the hypervisor attack vector for the four open-source platforms: *Proxomox VE, XenServer, KVM,* and *Oracle Virtual Box*. The detected vulnerabilities were rated based on their severity, with a focus on those with the most critical impact (Walkowski et al., 2021).

A vulnerability analysis was also performed using recursive penetration tests to assess virtual machine isolation levels and identify vulnerable source files, attack sources, and access methods. This is a high-coverage approach because text matching does not capture all vulnerabilities (Zheng & Li, 2024). Given that vulnerability analysis of open-source code can be performed in a dynamic (Ghelani et al., 2022) and real-time environment (Ghelani, 2022), a virtual test lab was set up with an Intel® CoreTM i7-8565 CPU@1.80GHz, a 1.99GHz x64-based processor, 16 GB of usable RAM, and a 64-bit operating system. The four hypervisors were installed as pre-built VMs to serve as attack targets.

Kali Linux is considered a holistic penetration test solution (Yarlagadda, 2024), with data capture from information gathering to exploitation of weak spots; therefore, the package was configured to run experiments concurrently (Ismaili, 2023; Nedyalkov, 2024) in both homogeneous and heterogeneous infrastructures. For clarity, each test was repeated three times, and vulnerabilities not directly related to hypervisors were treated as extraneous factors and so ignored and excluded from the analysis,

regardless of how they affected other layers of the virtual execution environment (Parast et al., 2022). These include, for example, security threats associated with the guest OS. The selected attack vectors are thought to be the most predominant issues that ideally affect hypervisor performance.

RESULTS AND FINDINGS

Hypervisor Attack Sources

Understanding attack sources and access methods is vital for assessing the security of virtual machines and their underlying hypervisors. The findings reveal that the sources of attack vectors vary between hypervisors. The attack vector in each hypervisor varied depending on whether the attack points were local or remote. Furthermore, attack vectors were found to originate from either hypervisor design faults or adopter misconfigurations. Regardless of these differences, each hypervisor was found with vulnerable source files. According to the vulnerability analysis, nearly two-thirds of the attacks (15) were caused by design faults and locally executed (16 attacks), as indicated in Table 2.

Table 2: Summary of Hypervisor Security Issues due to Attack Vectors

Hypervisor	Attack Vectors						Total
	Weak Point (Source)			Overall Attack Point			
	Configuration	Design	Both	Local	Remote	Both	
Proxmox VE	1	4	0	4	0	1	**10**
XenServer	1	4	0	3	1	1	**10**
KVM	2	3	0	4	1	0	**10**
OVB	2	4	0	5	1	0	**12**
Total	**6**	**15**	**0**	**16**	**3**	**2**	**42**

The results reveal that type 1 hypervisors are vulnerable to attacks that can occur both locally and remotely. Given the security state of all hypervisors, the Proxmox VE, XenServer, and KVM have slightly similar security strengths. On the other hand, OVB is more vulnerable than other hypervisors, with most attacks being caused by design flaws and locally executed.

While the clear design flaw in Proxmox VE is a failure in automatic RAM allocation, the XenServer suffers from modified file formats of the raw disk image in the guest VHD. For type 2, the most vulnerable part of KVM is the unsecured vCPU index in source files related to

arch/x86/kvm/, as well as the *core subcomponent*, *VDMA*, and *privilege escalation in guest machines* for OVB. Table 3 summarizes the effect of each attack vector on each hypervisor.

Table 3: Attack Vectors in each hypervisor

Attack Vectors	Open-Source Hypervisors				Total
	Type 1		Type 2		
	Proxmox VE	XenServer	KVM	OVB	
Soft MMU	Yes	Yes	Yes	Yes	**4**
Virtual CPUs	Yes	No	Yes	Yes	**3**
Interrupt and Timer Mechanism	Yes	Yes	Yes	No	**3**
I/O and Networking	Yes	Yes	Yes	No	**3**
Hypercalls	Yes	Yes	Yes	No	**3**
Remote Management Software	No	Yes	No	Yes	**2**
VM Exits	No	No	No	Yes	**1**
Hypervisor Add-ons	No	No	No	Yes	**1**
Symmetric Multiple Processor	No	No	No	No	**0**
Para-virtualized I/O	No	No	No	No	**0**
VM Management	No	No	No	No	**0**
Total	**5**	**5**	**5**	**5**	**20**

As shown in *Table 3*, all four hypervisors were found to be vulnerable to at least five attack vectors, with a soft MMU being the most prevalent. The interrupt and timer mechanisms, I/O and networking, and hypercalls all have an impact on Proxmox VE, XenServer, and KVM. The soft MMU in Proxmox VE is caused by a failure of automatic RAM allocation, whereas in XenServer, the main cause is the presence of vulnerable source files such as *include/asm-x86/debugreg.h* and *arch/x86/physdev.c*, which modify the file formats of the raw disk image in the guest VHD. The soft MMU in type 2 hypervisors is caused by the failures in both the core subcomponent and VDMA for OVB, as well as the presence of vulnerable source files such as *drivers/net/virtio_net.c* and *virt/kvm/ioapic.c* for KVM. In all hypervisors, the attack vector is locally executed due to design weaknesses.

Guest-to-Guest and Host-to-Guest Attacks

According to the analysis, all four hypervisors are vulnerable to guest-to-guest (G2G) and host-to-guest (H2G) attacks, whether performed locally or remotely, and whether caused by adopter misconfigurations or design

flaws. The G2G and H2G attacks help to determine the level of virtual machine isolation. Vulnerability analysis reveals the possibility of a host attack from the guest machine, resulting in the breakout of the underlying internal physical kernel isolation. Table 4 shows the attack possibilities for the G2G and H2G in each hypervisor.

Table 4: Summary of Hypervisor G2G and H2G Attacks

Attack Vectors	**Type 1**				**Type 2**				**Total (%)**
	Proxmox VE		**XenServer**		**KVM**		**OVB**		
	G2G	**H2G**	**G2G**	**H2G**	**G2G**	**H2G**	**G2G**	**H2G**	
Virtual CPUs Symmetric	√	√	X	X	√	√	√	√	**75**
Multiple Processors	X	X	X	X	X	X	X	X	**0**
Soft MMU	√	√	√	√	√	√	√	√	**100**
Interrupt and Timer Mechanism	√	√	√	√	√	√	X	X	**75**
I/O and Networking	√	√	√	√	√	√	X	X	**75**
Para-virtualized I/O	X	X	X	X	X	X	X	X	**0**
VM Exits	X	X	X	X	X	X	√	√	**25**
Hypercalls	√	√	√	√	√	√	X	X	**75**
VM Management	X	X	X	X	X	X	X	X	**0**
Remote Management Software	X	X	√	X	X	X	√	X	**25**
Hypervisor Add-ons	X	X	X	X	X	X	√	√	**25**
Total	**5**	**5**	**5**	**4**	**5**	**5**	**5**	**4**	

Analysis also shows that the rate of G2G attacks is consistent across all hypervisors, implying that they offer the same virtual machine isolation level. However, Proxmox VE and KVM are significantly more resistant to H2G attacks than XenServer and OVB. Fig 2 depicts G2G and H2G attacks across all hypervisors.

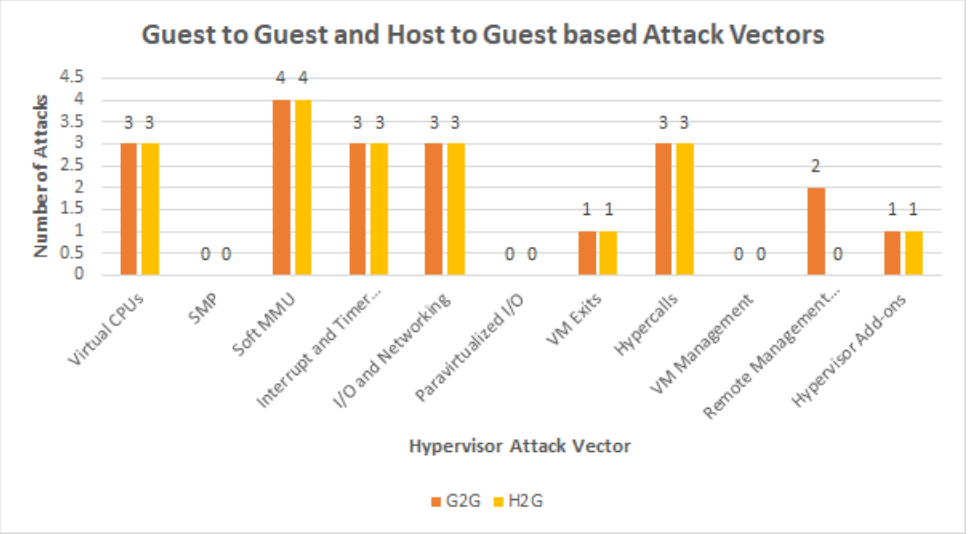

Figure. 2: G2G and H2G based on hypervisor attack vectors.

DISCUSSION

Essentially, all hypervisors are vulnerable to attack vectors, with the soft MMU being the most common. For attacks on computing resources such as virtual CPUs, XenServer was proven to be more secure than all other hypervisors. The result ties well with other studies that have reported optimal CPU performance in XenServer (Poojara et al., 2018). Unlike other hypervisors, the OVB is not vulnerable to interrupt and timer mechanisms, I/O and networking, and hypercalls, meaning that it provides better services for disk security and network performance.

All four hypervisors have shown a similar pattern of results with a possibility to break down into an attack on the host hypervisor from the guest machine. This is consistent with what has been found by Bazm *et al*. (2017), who claimed that distributed side-channel attacks are true for H2G attacks and have more weight in terms of isolation violation in the virtual environment. From these results, it is clear that attacks due to design faults occur on all hypervisors, regardless of type, with major parameters being virtualization method, maximum capacity of virtual resource allocation, software maturity status, and host and guest OS compatibility. On the other hand, common parameters with impacts on security configurations include the incorrect choice of virtual infrastructure between homogeneous and heterogeneous, preloaded and prebuilt hardware-specific and open-source drivers, resource allocation

mode between static and dynamic methods, vulnerable default configurations and patches, nesting, virtual machine states between dormant and active, and guest unavailability.

The prevalence of H2G attacks is also explained by Cheng et al. (2018) as the general breakout of physical kernel isolation. The study validates the two most infection layers (VM—VM and VM—OS) of the virtualized physical server, namely inter-virtual machine isolation (G2G) and VM-hypervisor isolation (H2G) (Cheng et al., 2018; Ara et al., 2020). The finding was quite surprising that all four hypervisors are not vulnerable to symmetric multiple processors, para-virtualized I/O, or VM management attacks, implying that not every attack vector can penetrate at the G2G and H2G levels.

While attacks on virtual CPUs are common in type 2 hypervisors, KVM adopters should pay close attention to interrupt and timer mechanisms, I/O and networking, and hypercalls throughout the configuration process, in line with KVM's large vulnerability quantity stated by Chen et al. (2023). OVB adopters should be keen on attacks related to VM exits, remote management, and hypervisor add-ons. When all four hypervisors are screened, G2G attacks outnumber H2G attacks, meaning that penetration attacks between virtual machines occur more frequently than between virtual machines and their underlying hypervisors. This implies that all hypervisors are more vulnerable to G2G attacks than H2G, although they offer the same isolation level against G2G attacks.

An important implication of these findings is that although type 1 hypervisors are more secure than type 2 hypervisors due to the security state and overhead factor of the host OS, any hypervisor type can be considered vulnerable to security threats if major design flaws and software misconfigurations are not properly addressed by software vendors and adopters. Thus, the findings are of direct practical relevance for adopters in ensuring that all essential security parameters are addressed throughout the deployment and adoption process, regardless of the chosen hypervisor type.

CONCLUSION

In conclusion, it is evident that hypervisor security is critical for adopters to attain maximum performance in virtual machines. While the shift from physical to virtual computing is constantly becoming popular with the use

of OSS solutions, the choice of an appropriate and secure hypervisor that is free from vulnerabilities and capable of responding to attack vectors is vital for an effective deployment process. This study has clearly shown the isolation strength between virtual machines for G2G and H2G attacks provided by the open source-based type 1 and type 2 hypervisors. The main conclusion that can be drawn is that all four hypervisors are vulnerable to security breaches, mostly due to design flaws and adapters' misconfigurations.

The study serves as a valuable resource for businesses and open-source adopters with a strategic plan to virtualize their computing resources. Further research should be conducted to explore isolation issues associated with container virtualization so that adopters can make informed decisions for the transition from physical to virtual computing.

RECOMMENDATIONS

Given the open-source nature of hypervisors and their widespread use in creating and managing virtual infrastructure, adopters should make significant efforts to overcome the possibility of G2G and H2G attacks, especially when open-source virtualization technology is used as a backend solution. Adopters should verify vulnerable source files on a regular basis, taking advantage of free access to open-source code. Throughout the deployment process, adopters should consider both software design and server configuration attributes, such as hypervisor upgrades, patching, and trusted software support sources. Adopters should also consider industry best practices, technology compliance and compatibility, feasibility studies, business process reviews, as well as a physical-to-virtual conversion plan.

REFERENCES

Aalam, Z., Kumar, V., & Gour, S. (2021). A review paper on hypervisor and virtual machine security. In *Journal of Physics: Conference Series* (Vol. 1950, No. 1, p. 012027). IOP Publishing.

Ally, S., Jiwaji, N. T., & Tarimo, C. (2018). A Review of Adopter's Common Misconfigurations of Virtual Machines: The Case of Tanzania. *Huria: Journal of the Open University of Tanzania*, *25*(2), pp. 158-180.

Ally, S. (2018). Comparative analysis of Proxmox VE and XenServer as type 1 open source based hypervisors. *International Journal of Scientific & Technology Research*, *7*(3), 72-77.

Anwer, M. B., Nayak, A., Feamster, N., & Liu, L. (2010). Network I/O fairness in virtual machines. In *Proceedings of the second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures* (pp. 73-80).

Ara, G., Lai, L., Cucinotta, T., Abeni, L., & Vitucci, C. (2020). A framework for comparative evaluation of high-performance virtualized networking mechanisms. In *International Conference on Cloud Computing and Services Science* (pp. 59-83). Cham: Springer International Publishing.

Arif, M., & Shakeel, H. (2015). Virtualization security: analysis and open challenges. *International Journal of Hybrid Information Technology*, *8*(2), 237-246, http://dx.doi.org/10.14257/ijhit.2015.8.2.22.

Basu, D., Wang, X., Hong, Y., Chen, H., & Bressan, S. (2019). Learn-as-you-go with megh: Efficient live migration of virtual machines. *IEEE Transactions on Parallel and Distributed Systems*, *30*(8), 1786-1801.

Bazm, M. M., Lacoste, M., Südholt, M., & Menaud, J. M. (2017). Side-Channels Beyond the Cloud Edge: New Isolation Threats and Solutions. In *IEEE International Conference on Cyber Security in Networking (CSNet)*, October *2017*.

Bridge, R., (2018). Open source software: Advantages & disadvantages. *Enterpreneur Handbook. https://entrepreneurhandbook.co.uk/open-source-software/, Date Accessed: 13-July-2018*

Chen, J., Li, D., Mi, Z., Liu, Y., Zang, B., Guan, H., & Chen, H. (2023). Security and Performance in the Delegated User-level Virtualization. In *17th USENIX Symposium on Operating Systems Design and Implementation (OSDI 23)* (pp. 209-226).

Cheng, Y., Zhang, Z., & Nepal, S. (2018). Still Hammerable and Exploitable: on the Effectiveness of Software-only Physical Kernel Isolation. *arXiv preprint arXiv:1802.07060*.

Chirammal, H. D., Mukhedkar, P., & Vettathu, A. (2016). *Mastering KVM virtualization*. Packt Publishing Ltd, 1st edition, ISBN 978-1-78439-905-4, Birmingham, UK, 2016.

Đorđević, B., Kraljević, N., & Kuk, K. (2024). Impact of Different Hypervisor Versions on the File System Performance: Case Study with VirtualBox. In *2024 23rd International Symposium INFOTEH-JAHORINA (INFOTEH)* (pp. 1-6). IEEE.

Duy, H. V., Xuan, T. N., Huyen, T. H. M., Dinh, T. N., Khanh, T. N., Quy, T. H., ... & Tran, T. (2024). Deploying Virtual Desktop

Infrastructure with Open-Source Platform for Higher Education. *Creative Approaches Towards Development of Computing and Multidisciplinary IT Solutions for Society*, 1-16.

Felter, W., Ferreira, A., Rajamony, R., & Rubio, J. (2015). An updated performance comparison of virtual machines and linux containers. In *2015 IEEE international symposium on performance analysis of systems and software (ISPASS)* (pp. 171-172). IEEE.

Freet, D., Agrawal, R., Walker, J. J., & Badr, Y. (2016). Open source cloud management platforms and hypervisor technologies: A review and comparison. *SoutheastCon 2016*, 1-8.

Gentile, A. F., Macrì, D., Greco, E., & Fazio, P. (2024). Overlay and Virtual Private Networks Security Performances Analysis with Open Source Infrastructure Deployment. *Future Internet*, *16*(8), 283.

Ghapanchi A. H., Aurum A., Low G. (2011). A taxonomy for measuring the success of open source software projects, First Monday, Vol. 16, No. 8

Ghelani, D., Hua, T. K., & Koduru, S. K. R. (2022). Cyber security threats, vulnerabilities, and security solutions models in banking. *Authorea Preprints*.

Ghelani, D. (2022). Cyber security, cyber threats, implications, and future perspectives: A Review. *Authorea Preprints*.

Goldman, R. (2016). *Learning Proxmox VE*. Packt Publishing Ltd.

Hirt T., (2010). KVM–the kernel-based virtual machine, 2010 *Red Hat Inc*.

Hoffmann, M., Nagle, F., & Zhou, Y. (2024). The Value of Open Source Software. *Harvard Business School Strategy Unit Working Paper*, (24-038).

Hyde, D. (2009). A Survey on the security of virtual machines. Security and Privacy, IEEE.

Iqbal, S., Kiah, M. L. M., Dhaghighi, B., Hussain, M., Khan, S., Khan, M. K., & Choo, K. K. R. (2016). On cloud security attacks: A taxonomy and intrusion detection and prevention as a service. *Journal of Network and Computer Applications*, *74*, 98-120.

Ismaili, M. A. (2023). Enhancing Cybersecurity: Exploring Effective Ethical Hacking Techniques with Kali Linux. *Research and Applications Towards Mathematics and Computer Science*, 135.

Kiszka, J., T DE IT, C. T., & Linux, C. C. C. E. (2010). Architecture of the kernel-based virtual machine (KVM). In *Siemens Availability:*

http://www. linux-kongress. org/2010/slides/KVM-Architecture-LK2010.pdf.

Kovari, A., & Dukan, P. (2012). KVM & OpenVZ virtualization based IaaS open source cloud virtualization platforms: OpenNode, Proxmox VE. In *2012 IEEE 10th Jubilee International Symposium on Intelligent Systems and Informatics* (pp. 335-339). IEEE.

Kulkarni, O., Bagul, S., Gawali, D., & Swamy, P. (2012). Virtualization technology: A leading edge. *International Journal of Computer Application*, 2(2). ISSN: 2250-1797

KVM., (2018). Kernel Virtual Machine - The Open Source Type 2 Hypervisor. https://www.linux-kvm.org/. Date Accessed: 22-10-2020.

KVM., (2021). Kernel Virtual Machine - The Open Source Type 2 Hypervisor" https://www.linux-kvm.org/, Date Accessed: 11-02-2021.

Liang, Z., Li, T., & Cui, E. (2024, May). RISC-V Virtualization: Exploring Virtualization in an Open Instruction Set Architecture. In *Proceedings of the 2024 5th International Conference on Computing, Networks and Internet of Things* (pp. 473-477).

Lozano, S., Lugo, T., & Carretero, J. (2023). A Comprehensive Survey on the Use of Hypervisors in Safety-Critical Systems. IEEE Access.

Mahjani, M. (2015). Security issues of virtualization in cloud computing environments. Master Thesis for award of Master of Science in Information Security degree at Lulea University of Technology, Sweden. 62pp.

MacDonald, M. (2013). Open Source Licensing in the Networked Era. *Masaryk University Journal of Law and Technology*, *7*(2), 229-239.

Masood, A., Sharif, M., Yasmin, M., & Raza, M. (2014). Virtualization tools and techniques: Survey. *Nepal Journal of Science and Technology*, *15*(2), 141-150.

Mishra, A. & Mishra, R. (2016). Virtualization security. *Global Research and Development Journal for Engineering (GRD)*, ISSN: 2455 – 5703, Vol. 1, Issue 12, pp. 20 – 24.

Mitre (2024). Mitre CVE Database, https://cve.mitre.org/. Date Accessed:11-03-2024.

Morabito, R., Cozzolino, V., Ding, A. Y., Beijar, N., & Ott, J. (2018). Consolidate IoT edge computing with lightweight virtualization. *IEEE Network*, *32*(1), 102-111.

Morabito, R., Kjällman, J., & Komu, M. (2015). Hypervisors vs. lightweight virtualization: a performance comparison. In *2015 IEEE*

International Conference on cloud engineering (pp. 386-393). IEEE.

Nazir, S. & Lazarides, M. (2016). Securing industrial control systems on a virtual platform: how to best protect the vital virtual business assets. White Paper, FIRSTCo

Nedyalkov, I. (2024). Study the Level of Network Security and Penetration Tests on Power Electronic Device. *Computers*, *13*(3), 81.

NIST (2021). NIST: National Institute of Standards and Technology. https://www.nist.gov/

NVD (2021). "National vulnerability database", https://nvd.nist.org, Date Accessed: 10-03-2021

NVD (2022). National vulnerability database. Website: https://nvd.nist.gov, Date Accessed: 11-12-2022

Obasuyi, G. C & Sari, A. (2015). Security challenges of virtualization hypervisors in virtualized hardware environment. International Journal of Communications, Network and System Sciences, 8(07), pp. 260-273, http://dx.doi.org/10.4236/ijcns.2015.87026

Pandey, R. (2020). Comparing vmware fusion, oracle virtualbox, parallels desktop implemented as type-2 hypervisors. *National College of Ireland*.

Panja, S. (2024). Information Retrieval Systems in Healthcare: Understanding Medical Data Through Text Analysis. In *Transformative Approaches to Patient Literacy and Healthcare Innovation* (pp. 180-200). IGI Global.

Parast, F. K., Sindhav, C., Nikam, S., Yekta, H. I., Kent, K. B., & Hakak, S. (2022). Cloud computing security: A survey of service-based models. *Computers & Security*, *114*, 102580.

Perez-Botero, D., Szefer, J., & Lee, R. B. (2013). Characterizing hypervisor vulnerabilities in cloud computing servers. In *Proceedings of the 2013 international workshop on Security in cloud computing* (pp. 3-10).

Poojara, S. R., Ghule, V. B., Birje, M. N., & Dharwadkar, N. V. (2018, December). Performance analysis of linux container and hypervisor for application deployment on clouds. In *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)* (pp. 24-29). IEEE.

Popek, G. J. & Goldberg, R. P. (1974). Formal requirements for virtualizable third generation architectures Communications of ACM, 17(7), pp. 412-421.

Proxmox VE Server Solutions (2016), https://www.proxmox.com/, software version 4.4, Date Accessed: 03-January-2017

Proxmox VE, (2018). The Open Source Type 1 Hypervisor, https://www.proxmox.com/en/proxmox-ve, 2018

Proxmox VE, (2021). The Open Source Type 1 Hypervisor, https://www.proxmox.com/en/proxmox-ve, Date Accessed: 16-05-2021

Rachana, S. C. & Guruprasad, H. S. (2014). Securing the virtual machines. *International Journal of Computer Technology & Applications* (IJCTA), ISSN: 2229 – 6093, 5(3), pp. 1012-1019

Ramana, V. V., Reddy, Y. S., Reddy, G. R. S., and Ravi, P. (2015). An assessment of virtual machineassails. *International Journal of Advanced Technology in Engineering and Science*, www.ijates.com, ISSN: 2348 – 7550, 03(01) pp. 315–320.

Reddy, N., Nadesh, R. K., Srinivasa Perumal, R., Mallela, N. C., & Arivuselvan, K. (2022). Performance Evaluation and Comparison of Hypervisors in a Multi-Cloud Environment. *Operationalizing Multi-Cloud Environments: Technologies, Tools and Use Cases*, 217-236.

Rodríguez-Haro, F., Freitag, F., Navarro, L., Hernánchez-sánchez, E., Farías-Mendoza, N., Guerrero-Ibáñez, J. A., & González-Potes, A. (2012). A summary of virtualization techniques. *Procedia Technology*, *3*, 267-272.

Singh, H., Manhas, P., Maan, D., & Sethi, N. (2016). Cloud computing security and privacy issues–A systematic review. *International Science Press*. IJCTA 9(11) 2016, pp. 4979–4992.

Singh, R., Kahlon, D., & Singh, S. (2016). Comparative Study of virtual machine migration techniques and challenges in Post Copy Live Virtual Machine Migration. *International Journal of Science and Research (IJSR) ISSN (Online)*, 2319-7064.

Sinha, V. A., & Thakare, V. M., (2023). Domain Analysis and Feature Identification of Virtualization in Debian Linux Using Type II Hypervisor–Virtual Box.

Stoyanov, B. (2024). Virtualization–concept and development. Virtualization environment and tools. *Business, New Technologies and Smart Society*, *2*(1), 9-25.

Thales, (2018). 2018 Global threat report, *451 Group for Thales, 2018 Thales Data Threat Report - Global Edition*, https://dtr.thalesesecurity.com/, Date Accessed: 13-July, 2018

Tiemann M., (2009). How open source software can save the ICT industry one trillion dollars per year. http://www.opensource.org/files/OSS-2010.pdf, Date Accessed: 13-July, 2018.

Tu, Q. (2000). Evolution in open source software: A case study. In *Proceedings 2000 International Conference on Software Maintenance* (pp. 131-142). IEEE.

Vainio, N., & Vadén, T. (2012). Free Software Philosophy and Open Source. *International Journal of Open Source Software and Processes (IJOSSP)*, *4*(4), pp. 56-66.

Verizon, (2017). 2017 Data breach investigations report, 10th Edition, https://www.ictsecuritymagazine.com/wp-content/uploads/2017-Data-Breach-Investigations-Report.pdf, Date Accessed: July 13, 2018

Virtualbox, O. V. (2011). Oracle vm virtualbox. *Change*, *107*, pp. 1-287.

Virtualbox, O. (2018). Oracle Virtual Box - The Open Source Type 2 Hypervisor. https://www.virtualbox.org, 2018

Virtualbox, O. (2021). Oracle Virtual Box - The Open Source Type 2 Hypervisor. https://www.virtualbox.org, Date Accessed: 13-05-2021

Vojnak, D. T., Đorđević, B. S., Timčenko, V. V., & Štrbac, S. M. (2019). Performance Comparison of the type-2 hypervisor VirtualBox and VMWare Workstation. In *2019 27th Telecommunications Forum (TELFOR)* (pp. 1-4). IEEE.

Vojtesek, J., & Pipis, M. (2016). Virtualization of operating system using type-2 Hypervisor. In *International Conference on Information and Communication Technology for Intelligent Systems* (pp. 239-247). Cham: Springer International Publishing.

Walkowski, M., Oko, J., & Sujecki, S. (2021). Vulnerability management models using a common vulnerability scoring system. *Applied Sciences*, *11*(18), 8735.

Wueest, C. (2014). Threats to virtual environments. Security Response, Symantec, version 1, pp.18

XenServer Hypervisor, (2017). Xen Hypervisor-Software version 4.8, https://xenproject.org/, Date Accessed: 04-January-2017

XenServer, (2018). The Open Source Type 1 Hypervisor https://xenserver.org, 2018

Yarlagadda, V. (2024). Harnessing Kali Linux for Advanced Penetration Testing and Cybersecurity Threat Mitigation. *Journal of Computing and Digital Technologies*, *2*(1), 22-35.

Yauri, B. A., & Abah, J. (2016). Mitigating security threats in virtualized environments. *International Journal of Computer Science and Network Security (IJCSNS)*, 16(1), pp. 101

Zajdel, S., Costa, D. E., & Mili, H. (2022). Open source software: an approach to controlling usage and risk in application ecosystems. In *Proceedings of the 26th ACM International Systems and Software Product Line Conference-Volume A* (pp. 154-163).

Zhang, Z., Liu, Y., Chen, J., Qi, Z., Zhang, Y., & Liu, H. (2021). Performance Analysis of Open-Source Hypervisors for Automotive Systems. In *2021 IEEE 27th International Conference on Parallel and Distributed Systems (ICPADS)* (pp. 530-537). IEEE.

Zhang, Z., Xia, C., Liang, C., Li, J., Yu, C., Bie, T., ... & Guan, H. (2024). Un-IOV: Achieving Bare-metal Level I/O Virtualization Performance for Cloud Usage with Migratability, Scalability and Transparency. *IEEE Transactions on Computers*.

Zheng, K., & Li, Z. (2024). An Image-Text Matching Method for Multi-Modal Robots. *Journal of Organizational and End User Computing (JOEUC)*, *36*(1), 1-21.

Zoughbi, D., & Dutta, N. Hypervisor Vulnerabilities and Some Defense Mechanisms, in Cloud Computing Environment. *International Journal of Innovative Technology and Exploring Engineering*, *10*, 42-48.