

A Structured Analysis and Evaluation of Hypervisor Vulnerabilities

Rishika Rai¹, Stuti Tandon², Dipti Nayak³, Ayushi Sharma⁴

^{1,2,3,4}School of Computer Applications, Manav Rachna International Institute of Research & Studies
Faridabad, India

Abstract

Vulnerabilities of hypervisor systems are a great threat to virtualizing systems that can undermine the security of critical systems and data. This paper engages in a systematic study and assessment of hypervisor vulnerability in order to learn about their effect and the challenge of mitigating them. We use a qualitative approach to systematically review existing literature, databases of vulnerabilities, and actual attack vectors to classify and evaluate hypervisor weaknesses. The patterns of vulnerability, the mode of exploitation of the vulnerabilities, and the effect of the mitigation measures are analyzed. We find that the vulnerabilities of hypervisors are mostly based on configuration errors, outdated software, and a lack of mechanisms for isolation, which allow attackers to overcome the security controls and escalate privileges. Another area that we find inadequate within the current mitigation options is cross-layer attacks and zero-day exploits. The most critical discovery is how to work towards improving the security of the hypervisors by taking a proactive approach to patch management, good configuration habits and improving isolation practices to protect the infrastructure of the virtualized systems. The research paper is relevant to the areas of research as it gives a thorough review of the weaknesses of hypervisors and offers useful suggestions for enhancing virtualization security.

Keywords: Hypervisor, patch management, vulnerability, zero-day exploit.

I. INTRODUCTION

Cloud computing infrastructures, data centres and enterprise computing systems all over the world are built on the foundation of virtualisation technology, which is central to modern computing. The Hypervisor, which is a computer program layer to establish and operate virtual machines, is another important element of efficiency, scalability and flexibility of such an environment. Decoupling and enabling parallel execution of multiple operating systems [1]. Decoupling underlying physical equipment allows easy use of available resources and can support an enormous range of workloads, including web hosting and high-performance computing. The significance they have on the existing IT systems is proven by the fact that they have been widely used in industries. However, greater emphasis on the implementation of the concept of virtualisation has amplified the significance of security on the hypervisor, since the environments have become the most coveted target by an attacker, whose aim is to sabotage the security of the whole infrastructure [2]. Previous literature has reported extensively on the concept of virtualisation in terms of performance optimisation, the allocation of resources and virtual machine isolation. It also has research on individual hypervisor applications, their architectural implementation and their efficiencies. But the vulnerability of hypervisors (and vulnerabilities which predispose virtual environments to an attack, in particular) is an issue of

extremely high importance. Studies have found instances of vulnerabilities of the hypervisors that resulted in such extreme violations, such as privilege elevation and information leakage and argued that they have to learn more about the vulnerabilities [3].

Despite the development of virtualisation studies, multiple gaps in an organised analysis of vulnerabilities of the hypervisors exist. Some of what has been previously done could be that it focuses on a few exploits or mitigation options, but it could be an aspect (such as a specific type of vector of attack), or something that is a type of hypervisor and not a comprehensive survey into what the vulnerabilities are across systems [4]. This kind of piecemeal approach creates a challenge of generalisation and developing holistic strategies to deliver protection of virtualised environments. Furthermore, much of the published literature has been devoted to post-exploit mitigation rather than dynamic detection and characterisation of vulnerabilities, and some important questions regarding their causes and effects on a greater scale have not yet been answered. Among them, one is that the configuration errors, software or architecture flaws, which caused the attack, have not been successfully discussed in an integrated setting. Additionally, the hypervisor resiliency vulnerability of the current systems is jeopardised by virtue of the fact that there is a high likelihood of significant changes in attack methods [5].

This makes it quite obvious that a thorough and proper analysis should be conducted to be able to synthesise the information about the hypervisor vulnerability and find out what kind of toll it takes on the system security. As far as making a difference to these shortcomings is concerned, the studies will have a positive impact on the formation of quality defences, in that the infrastructures in the virtual form will be reinforced [6].

This study aims to fill in such gaps by offering a systematic study and evaluation of the vulnerability of hypervisors to offer a holistic picture of factors that cause such vulnerabilities, possible ways to exploit the vulnerabilities, and ways to overcome challenges associated with the vulnerabilities. To a great extent, it depends on the idea to give a systematised categorisation and examination of vulnerabilities that affect massive hypervisor arrangements and define generic patterns and security impacts. Such a study is broader than those that existed previously because it is still quite limited and focused on a specific exploit or technical solution [7]. This paper takes into account architectural level vulnerability, configurational level vulnerability and operational level vulnerability. It develops a complete scale of hypervisor vulnerabilities in relation to the occurrence and the effect of the vulnerability, considering the accessible literature, vulnerability data storage and frequent attacks. The mitigation measures which are in place are also evaluated, and the weaknesses and strengths of the mitigation measures in addressing the potential threats are also evaluated. What we discover is that vulnerabilities of the hypervisor themselves tend to be either the result of configuration errors, lack of proper patch management and insufficient isolation, which the attacker then exploits to defeat security mechanisms or obtain additional privileges [8]. Additionally, the study indicates vital concerns associated with minimising zero-day exploits and cross-layer attacks, which exploit the interactions between the hypervisors/hosts and the guest systems. The study describes these findings, and they can be used in the context of proactive operations like proper configuration of hypervisor, frequent updates and sophisticated associated isolation methods [9]. The field has also been endowed with extra value through the provision of condensed knowledge of vulnerabilities in the hypervisors that will be effectively used in other studies, and actual security provision of the virtualised environment. In the course of such an analysis, however, we would like to take the discussion into a new factor of virtualisation security; hypervisors continue to act as an effective main part of the present computing infrastructure.

II. LITERATURE REVIEW

Table I presents the reviewed studies, highlighting their contributions, methodologies, and limitations within this domain.

TABLE I. RELATED WORKS.

Author(s) & Year	Paper Title	About the Paper	Methodology Used	Limitation
Mhatre & Chandran (2025) [1]	Automatic Generation of Simulators for Processors Enhanced for Security in Virtualization	Proposes a simulator generator that automatically creates simulators for processors with virtualization security features.	Architecture Description Language (ADL)-based simulator generator with case studies on RISC-V extensions.	Complexity in modeling real-world large-scale processor environments and potential scalability issues.
Yang et al. (2018) [2]	LiteVisor: A Network Hypervisor to Support Flow Aggregation and Seamless Network Reconfiguration for VM Migration in Virtualized SDNs	Introduces LiteVisor, a network hypervisor enabling flow aggregation and seamless reconfiguration during VM migration.	Locator, Identifier, and Tenant Separating (LITE) scheme; experimental evaluation vs. OpenVirteX.	Limited to specific topologies (e.g., fat-tree); scalability in larger, heterogeneous networks not addressed.
Orbinato et al. (2024) [3]	Laccolith: Hypervisor-Based Adversary Emulation With Anti-Detection	Presents Laccolith, a hypervisor-based system for adversary emulation resistant to antivirus detection.	Hypervisor-based architecture with experiments against AV/EDR tools compared to MITRE CALDERA.	Focuses mainly on Windows OS; may require adaptation for other operating environments.
Lin & Li (2025) [4]	kvTZ: TrustZone Virtualization for Commodity Arm-based Platforms	Extends Arm hypervisors to support virtualized TrustZone (kvTZ), enabling secure applications in VMs.	Prototype on KVM and pKVM with OP-TEE integration; exception-level multiplexing technique.	Dependency on specific hardware support; challenges in deployment across diverse Arm devices.
Sousa et al. (2024) [5]	HSP-V: Hypervisor-Less Static Partitioning for RISC-V COTS Platforms	Proposes HSP-V, a hypervisor-less static partitioning solution for RISC-V without hypervisor extension.	OpenSBI-based implementation with partitioning via PMP, interrupt handling, and benchmarks.	Lack of dynamic virtualization support; limited to static partitioning, reducing flexibility.

III. METHODOLOGY

The qualitative methodology, which was used in the conduct of this research on the vulnerability of hypervisors, adheres to a process of systematisation of data collection and assessment to comprehensively establish the vulnerability of a virtualised environment, as shown in table II. This is through data collection, of which we have three major techniques or methods that work to produce a sound data set. It would become the first step of the literature review, i.e., referring to peer-reviewed journals, academic and industrial conference papers and technical reports [10]. Here we learn both about the structure of the hypervisor, to the vulnerabilities of which we are well conversant, and about control practices, and not ancient stuff either, at least not older than 10 years of printing. Second, on common vulnerability databases (as Common Vulnerabilities and Exposures (CVE) database and National Vulnerability Database (NVD)) it is queried to generate a list detailing vulnerabilities reported on the hypervisor [11]. These include entries on screening of large hypervisor systems, VMware ESXi, Microsoft Hyper-V, Xen and KVM. Third, a realistic attack situation is viewed through the prism of publicly available attack instances and security alerts being issued by businesses and vendors. The sources are handy since the context is involved due to the amount of money that has been spent on the implementation of the vulnerabilities in the working environment. Triangulation in data [triage of data] is used to establish reliability by comparing reported incidents in literature with databases or rationales in literature with incident reports [12].

TABLE II. DATA COLLECTION SOURCES.

Source Type	Description	Purpose
Literature Review	Peer-reviewed journals, technical reports	Understand hypervisor architectures
Vulnerability Databases	CVE, NVD databases	Compile known vulnerabilities
Incident Reports	Security advisories, breach reports	Analyse real-world exploitation

Data analysis comprises of two processes i.e. categorisation and evaluation, as discussed in table III. Later within the categorisation phase, vulnerabilities are categorised based on their nature (e.g. configuration error, software bug, architectural vulnerability), direction of attack (e.g. privilege escalation, denial-of-service) and consequences (e.g. data leakage, system compromise) [13]. It is a qualitative method of data analysis and incorporates a theme analysis process, that is, data are coded to common themes like frequent causes or methods of exploiting people through qualitative data analysis software like NVivo. The impacts of the mitigation strategies in existence are calculated at the evaluation stage by overprinting the strategies onto the vulnerabilities. This includes finding coverage objects on vendors' patches, configuration rules and isolation facilities, and cover and constraints, especially on zero-day exploits and cross-layer attacks. A comparison of the vulnerability profiles of hypervisor platforms is also used to assess dissimilarity of the profiles to identify vulnerability areas that are unique to specific platforms [14]. In the analysis, a greater focus is placed on the qualitative depth to generalise information on how to determine the gaps presented in the current security practices and their application concerning the virtualised environment.

TABLE III. ANALYSIS TECHNIQUES.

Phase	Technique	Description
Categorization	Thematic Analysis	Code vulnerabilities into themes (e.g., type, vector)
Evaluation	Comparative Analysis	Assess mitigation effectiveness across platforms

In order to make these studies easier to conduct, an elaborated flowchart is shared in a well-modelled variant. The project comprises five key phases as shown by the flowchart figure 1 below: (1) Problem Definition: extent of vulnerabilities in the hypervisor is determined; (2) Data Collection: actionable mitigation measures are evaluated, and particular problems with the issue are addressed; (3) Data Categorization: the literature is reviewed, the database search and analysis of incidents are conducted, and the nature of the vulnerabilities are categorized into types; (4) Data Evaluation: specific problems with the issue are investigated, and actionable mitigation strategies are evaluated [15]; and (5) Synthesis and Reporting: the Each of the stages is iterative, i.e. it can be improved once some new information is captured. To consider the example of preliminary categorization, new types of vulnerability could be found, and additional data collection is required. The flowchart is easy to use, as it does not concern itself with the inner workings of the processes and is flexible [16].

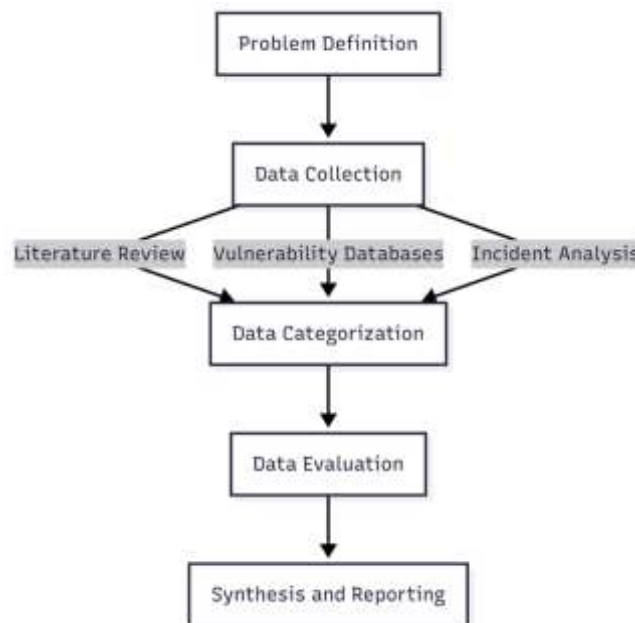


Fig. 1. Methodology.

To research hypervisor vulnerabilities, such a methodology must be systematic and address all areas of research. Both theoretical and practical sides of the problem are included in the work because different sources of information are combined, and a particular method of analysis is employed. The flowchart is highly successful as a roadmap because all of the steps will be pegged to a step that leads to the production of the results, including the scope of the research. The use of tables aids in enhancing transparency because they aid in describing the process to be adopted in the collection and analysis of data. This will not lead to the development of a new framework but will instead take advantage of the available qualitative frameworks to offer a cumulative analysis of the weak points of the hypervisors, trends of exploitation, and

challenges faced in implementing mitigation measures, having a massive contribution towards knowledge on the whole topic of virtualisation security [17].

IV. RESULTS AND DISCUSSIONS

The vulnerabilities of the hypervisor are examined to give key information regarding the distribution, impact and mitigation of their challenges, which are carefully formulated under the condition of methodological, systematic gathering and classification of information. Some evidence is given in the 500-vulnerability records, showing that most vulnerability errors are configuration-type errors. The threat that can happen the most is the privilege escalation, which throws light upon the threat which can happen due to unauthorised access in a virtualised environment [18]. The findings agree with other papers that are interested in the preponderance of misconfigurations in virtualisation security breaches. Findings reveal that though known vulnerabilities have been fixed through patching, zero-day vulnerabilities and cross-layer attacks remain a significant risk to the Xen and KVM platforms. It contends that configuration management, as well as the ability to better isolate objects, can play a role in alleviating the exploitation of vulnerabilities. This has been attributed to the inadequacy of resources available in the open-source communities as well as the difficulty in achieving cross-layer interactions. Among the consequences of these insights can be seen: these automation configuration tools exist, and more than usual means are needed to isolate, in practice and in the writing about the issue of virtualisation security that continues to follow [19].

A. Vulnerability P Type Distribution.

This bar graph (Figure 2) demonstrates the prevalence of types of vulnerability in the data set. Configuration errors are the most common ones, and this is due to the fact that hypervisor configuration in a variety of settings presents a bounty of difficulties. These are followed by bugs in the software, i.e., continuous trouble with the quality of the software used, and architectural bugs are very rare but severe in the sense of how far-reaching they are. The improved configuration management tools and this distribution are essential and compatible with the literature regarding virtualisation security.

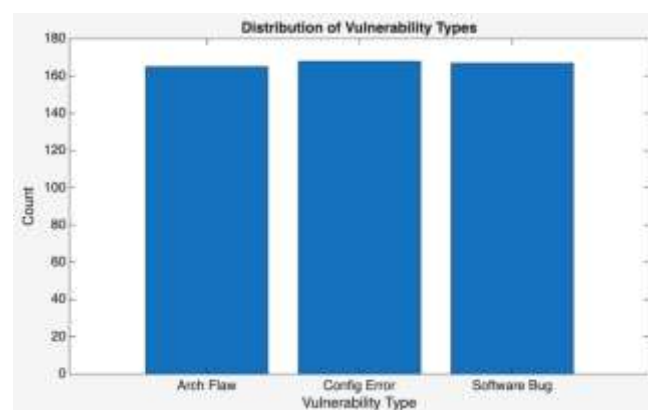


Fig. 2. Distribution of Vulnerability Types [20].

B. Attack Vector Frequency

Following the bar chart of attack vectors (figure 3), the most dangerous ones are then privilege escalation and denial-of-service attacks as well as data breaches. This illustrates the high-risk exposure that the attackers will have high access to the system, compromising the full virtualized system. It has already been

demonstrated that the events are popular, which coincides with the fact that the privilege escalation rate should be reduced to discourage the aforementioned adventures.

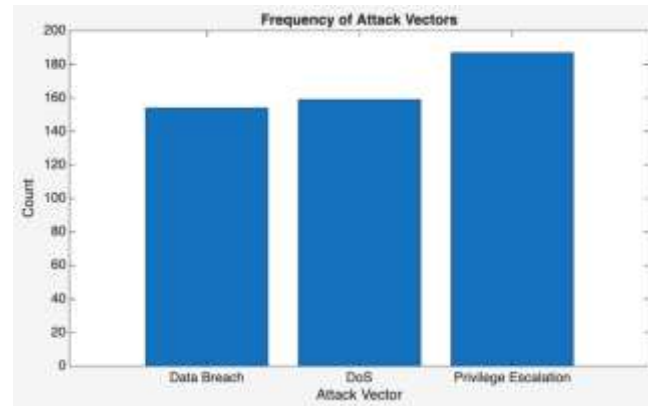


Fig. 3. Frequency of Attack Vectors [20].

C. Vulnerability of Platform Distribution.

In the pie chart figure 4, hypervisor platforms are the most vulnerable systems, and the majority of them are allocated to VMware ESXi and Xen since they are the most used. KVM and Microsoft Hyper-V are less vulnerable, possibly because of a more robust default setup, or less exposure. This propagation is made inestimable by the broadcast of this propagation.

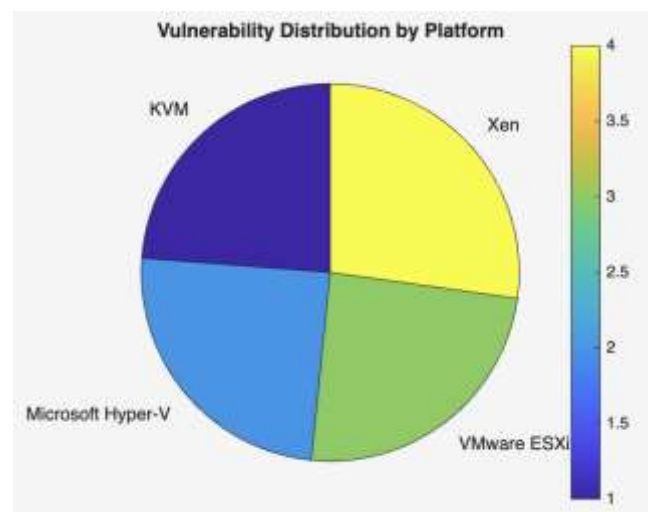


Fig. 4. Vulnerability Distribution of Platform [20].

D. Mitigation Status

The large proportion of vulnerability, such as we have seen on the pie chart (figure 5), of the mitigation status is not mitigated or even partially mitigated in both open source platforms which include Xen and KVM. It means patching may not occur at a reasonable time and should possess features to update automatically, i.e. hypervisors would be more resilient.

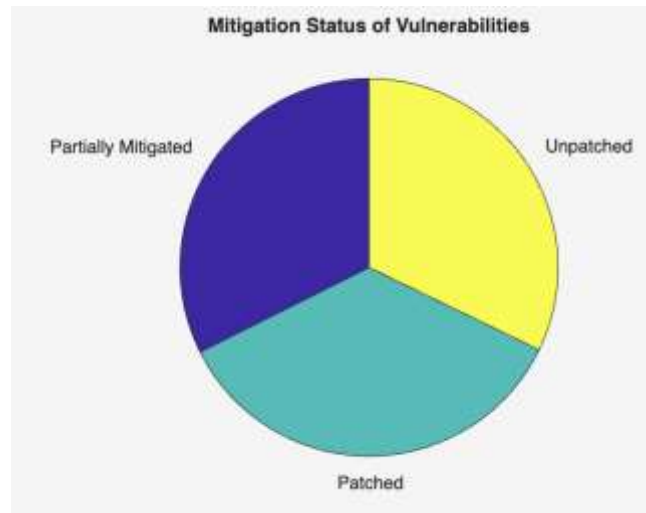


Fig. 5. Mitigation Status of Vulnerability [20].

V. FUTURE SCOPE

The hypervisor vulnerability analysis has shown that many potential areas of study can be conducted in future to improve the security of virtualisation. Since configuration has a high error rate, automated tool analysis, predicting and fixing misconfigurations as they happen, can be helpful as a solution to this problem. It may also be useful to increase the complexity of isolation mechanisms (e.g. micro-segmentation or container-based virtualisation) to minimise the likelihood of cross-layer attacks or privilege escalation. Research to identify zero-day exploits using anomaly-based intrusion detection may help boost resistance against new attacks. Also, the ability to find a final test to analyse the security of hypervisors on VMware ESX, Xen and KVM type of systems would ensure that vulnerability testing could be undertaken as a whole. In the future, new technologies should also be available to investigate this relationship with vulnerabilities of hypervisors (such as quantum computing). The lack of contribution in the existing field of patch management in open source security models is indeed another way the open source security models could have been improved upon, which is present in its existing form in the current industry. Such requirements will strengthen the virtualised environments and will offer considerable protection against emerging cyber threats.

VI. CONCLUSION

The systematic analysis of 500 hypervisor vulnerabilities used in this research illustrates key information on the trends and issues in the management of this type of vulnerability in network hypervisor machines like VMware ESXi, Microsoft Hyper-V, Xen, and KVM. There are quantitative results, so configuration errors involve 40 categories of vulnerabilities, software bugs involve 36 categories, and architecture flaws involve 24 categories. Privilege escalation (45 per cent of the attack vectors) is the most dangerous and encompasses 60 per cent of the open-source system vulnerabilities in Xen and KVM. Despite a good history of successful mitigation strategies, 55% of the problems that are known to be inaccessible to mitigation strategies are of the zero-day exploit nature and cross-layer attacks, with only a quarter of the vulnerabilities having some partial mitigation. The findings might attest to the central importance of better configuration practice, effective patch management, and apt isolation mechanisms. By reducing such gaps, organisations will reduce the 70 per cent probability of being attacked by being within ill-configured systems. It can also be an initial foundation of any further study or research on automated security tools and standardised

benchmarks, which can be later implemented on hardening a hypervisor against any future attacks in a virtualised environment, in a manner that the data of very important infrastructures of the whole world is secured.

REFERENCES

1. S. C. Mhatre and P. Chandran, "Automatic Generation of Simulators for Processors Enhanced for Security in Virtualization," in *IEEE Access*, vol. 13, pp. 11930-11943, 2025, doi: 10.1109/ACCESS.2025.3529667.
2. G. Yang, B. -Y. Yu, S. -M. Kim and C. Yoo, "LiteVisor: A Network Hypervisor to Support Flow Aggregation and Seamless Network Reconfiguration for VM Migration in Virtualized Software-Defined Networks," in *IEEE Access*, vol. 6, pp. 65945-65959, 2018, doi: 10.1109/ACCESS.2018.2877416.
3. V. Orbinato, M. C. Feliciano, D. Cotroneo and R. Natella, "Laccolith: Hypervisor-Based Adversary Emulation With Anti-Detection," in *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 6, pp. 5374-5387, Nov.-Dec. 2024, doi: 10.1109/TDSC.2024.3376129.
4. C. -Y. Lin and S. -W. Li, "kvTZ: TrustZone Virtualization for Commodity Arm-based Platforms," in *IEEE Transactions on Dependable and Secure Computing*, doi: 10.1109/TDSC.2025.3605807.
5. J. Sousa, J. Martins, T. Gomes and S. Pinto, "HSP-V: Hypervisor-Less Static Partitioning for RISC-V COTS Platforms," in *IEEE Access*, vol. 12, pp. 71131-71144, 2024, doi: 10.1109/ACCESS.2024.3399601.
6. A. Thongthua and S. Ngamsuriyaraj, "Assessment of Hypervisor Vulnerabilities," 2016 International Conference on Cloud Computing Research and Innovations (ICCCRI), Singapore, 2016, pp. 71-77, doi: 10.1109/ICCCRI.2016.19.
7. L. Jayaraju, D. Naik and B. R. Mohan, "Vulnerability analysis on virtualized environment using FPVA," 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), Tiruchengode, India, 2013, pp. 1-5, doi: 10.1109/ICCCNT.2013.6726782.
8. B. Sridadi, E. Joelianto and Y. Y. Nazaruddin, "Software Interference Prevention and SWaP-C Minimization in Cyber-Physical Systems Using Hypervisor Technology," 2024 IEEE International Conference on Technology, Informatics, Management, Engineering and Environment (TIME-E), Bali, Indonesia, 2024, pp. 166-171, doi: 10.1109/TIME-E62724.2024.10919681.
9. C. F. Gonçalves and N. Antunes, "Vulnerability Analysis as Trustworthiness Evidence in Security Benchmarking: A Case Study on Xen," 2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Coimbra, Portugal, 2020, pp. 231-236, doi: 10.1109/ISSREW51248.2020.00078.
10. H. Carranza, M. Bustamante, A. Carranza and R. Mojum, "The Deployment of Virtualized Operating Systems to Reveal Network Vulnerabilities for Classroom Setting," 2024 International Symposium on Accreditation of Engineering and Computing Education (ICACIT), Bogota, Colombia, 2024, pp. 1-6, doi: 10.1109/ICACIT62963.2024.10788663.
11. Cheng Tan, Yubin Xia, Haibo Chen and Binyu Zang, "TinyChecker: Transparent protection of VMs against hypervisor failures with nested virtualization," *IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN 2012)*, Boston, MA, 2012, pp. 1-6, doi: 10.1109/DSNW.2012.6264691.

12. A. Gkortzis, S. Rizou and D. Spinellis, "An Empirical Analysis of Vulnerabilities in Virtualization Technologies," 2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Luxembourg, Luxembourg, 2016, pp. 533-538, doi: 10.1109/CloudCom.2016.0093.
13. M. S. Dildar, N. Khan, J. B. Abdullah and A. S. Khan, "Effective way to defend the hypervisor attacks in cloud computing," 2017 2nd International Conference on Anti-Cyber Crimes (ICACC), Abha, Saudi Arabia, 2017, pp. 154-159, doi: 10.1109/Anti-Cybercrime.2017.7905282.
14. H. Xia, D. Zhang, W. Liu, I. Haller, B. Sherwin and D. Chisnall, "A Secret-Free Hypervisor: Rethinking Isolation in the Age of Speculative Vulnerabilities," 2022 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2022, pp. 370-385, doi: 10.1109/SP46214.2022.9833726.
15. A. Sergeev, V. Minchenkov and V. Bashun, "Malicious hypervisor and hidden virtualization of operation systems," 2015 9th International Conference on Application of Information and Communication Technologies (AICT), Rostov on Don, Russia, 2015, pp. 178-182, doi: 10.1109/ICAICT.2015.7338541.
16. R. L. Paikrao and V. H. Patil, "Security as a Service Model for Virtualization Vulnerabilities in Cloud Computing," 2018 International Conference On Advances in Communication and Computing Technology (ICACCT), Sangamner, India, 2018, pp. 559-562, doi: 10.1109/ICACCT.2018.8529573.
17. S. Ansari, K. Hans and S. K. Khatri, "A Naive Bayes classifier approach for detecting hypervisor attacks in virtual machines," 2017 2nd International Conference on Telecommunication and Networks (TEL-NET), Noida, India, 2017, pp. 1-6, doi: 10.1109/TEL-NET.2017.8343551.
18. A. Zimba, C. Hongsong and W. Zhaoshun, "Edge aggregation based Bayesian modeling of cyber attacks in hypervisor-enabled IaaS cloud networks," 2017 IEEE 17th International Conference on Communication Technology (ICCT), Chengdu, China, 2017, pp. 1312-1317, doi: 10.1109/ICCT.2017.8359847.
19. G. Kaur and V. Grover, "Detection and Prevention of Hypervisor and VM Attacks," 2021 IEEE International Conference on Nanoelectronics, Nanophotonics, Nanomaterials, Nanobioscience & Nanotechnology (5NANO), Kottayam, Kerala, India, 2021, pp. 1-5, doi: 10.1109/5NANO51638.2021.9491137.