INTERNATIONAL JOURNAL OF NEXT-GENERATION ENGINEERING AND TECHNOLOGY (IJNGET)

eISSN: 3087-4319

Volume. 02, Issue. 10, pp. 09-16, October2025

Securing Multi-Tenant Cloud Environments: Architectural, Operational, and Defensive Strategies Integrating Containerization, Virtualization, and Intrusion Controls

Dr. Elena M. Carter
University of Edinburgh

Article received: 01/10/2025, Article Accepted: 17/10/2025, Article Published: 31/10/2025

ABSTRACT

This paper presents a comprehensive, publication-ready analysis of security architectures and operational strategies for multi-tenant cloud environments, synthesizing technical, organizational, and theoretical perspectives derived from the supplied literature. The investigation centers on tensions and complementarities between containerization and virtual machine paradigms, native multi-tenancy design considerations, intrusion detection and prevention mechanisms, and specialized applications within healthcare and distributed hospital environments. The work explicates a layered threat model for multi-tenant clouds that accounts for co-tenancy risks, resource isolation failures, orchestration vulnerabilities, and adversarial patterns including distributed denial-of-service (DDoS) campaigns and stealthy coordination attacks. Methodologically, the paper develops a descriptive, theory-driven framework for evaluating secure deployment choices—contrasting Docker containers and virtual machines (VMs) in terms of attack surface, resource isolation, operational agility, and security management overhead—while integrating multi-party computation as a privacy-preserving collaboration technique for sensitive data (e.g., healthcare) and mapping IDS/IPS capabilities to host- and network-level defenses. Results are presented as a set of synthesized findings: best-practice architectural patterns for native multi-tenancy, a taxonomy of intrusion detection/prevention duties across layers, recommended orchestration hygiene and configuration hardening steps for OpenStack and multi-node deployments, and a risk-prioritized set of controls for healthcare cloud systems. The discussion explores the theoretical implications for cloud security research, articulates limitations rooted in the constrained reference base, and outlines a future research agenda including empirical validation, automated vulnerability discovery in multi-tenant orchestration platforms, and integration of secure multi-party computation for cross-institutional health data sharing. This article delivers a dense, citation-anchored resource for researchers and practitioners seeking a holistic approach to securing multi-tenant cloud infrastructures.

Keywords: multi-tenant cloud security; container vs virtual machine; intrusion detection and prevention; OpenStack; secure multi-party computation; healthcare cloud; orchestration hardening

INTRODUCTION

The architecture and operation of contemporary cloud systems routinely assume a multi-tenant model wherein multiple independent customers share compute, storage, and networking resources while expecting logical separation and confidentiality (Guo et al., 2017). Multi-tenancy provides economic and operational benefits—resource pooling, elastic scaling, and simplified management—but simultaneously introduces complex security trade-offs that manifest at architecture, platform, orchestration, and application layers (Almorsy et al., 2017). Central to these trade-offs is the choice between virtualization technologies (virtual machines) and containerization technologies (e.g., Docker), each introducing distinct isolation semantics, management models, and risk profiles (AquaSec, 2019). Another persistent axis of concern is intrusion detection and prevention: decisions about whether to apply host-based or network-based mechanisms, how to place these controls in an environment of dynamic resource mobility, and how to scale detection capabilities without undermining tenant privacy (Ashoor & Gore, 2011; Singh & Singh, 2014).

The problem addressed in this paper is multifaceted and practical: how should architects and security engineers design multi-tenant cloud deployments—particularly those orchestrated by platforms such as OpenStack—so that they achieve the economic and functional benefits of sharing while minimizing cross-tenant security risk? The literature supplied frames core components of this question: comparative analyses of Docker containers and VMs (AquaSec, 2019), OpenStack configuration and multi-node deployment experience (Sehgal, 2012; Oracle, 2019), empirical frameworks for native multi-tenancy application design and management (Guo et al., 2017), and operational detection frameworks intended to actively detect vulnerabilities in multi-tenant systems (Flood & Keane, 2017). Implications for sensitive sectors—especially healthcare—are discussed through applied studies advocating secure multi-party computation and cloud data collection strategies for patient data (Marwan et al., 2017; Dharmaraju et al., 2016). Additionally, adversarial phenomena such as stealthy denial-of-service orchestration are documented in the domain literature and highlight the unique threat vectors that cloud orchestration layers must account for (Goutham & Tejaswini, 2016).

Despite prior work, a gap remains in the literature for an integrated, operationally actionable framework that: (a) situates container and VM decision-making within a broader multi-tenant risk taxonomy, (b) maps intrusion detection and prevention functions to the architectural layers of modern cloud platforms, and (c) recommends concrete orchestration and configuration practices for multi-node OpenStack deployments and other common cloud fabrications. This article addresses that gap by synthesizing the provided references into a theoretical and descriptive framework intended to be directly useful to lead researchers, cloud architects, and security operations teams. The focus remains strictly within the supplied reference base, leveraging each cited contribution to build an internally consistent argument for layered security, rigorous orchestration hygiene, and privacy-aware collaboration techniques for sensitive domains such as healthcare.

METHODOLOGY

The methodological approach of this study is intentionally descriptive, integrative, and theory-driven: rather than performing primary empirical measurements or new experimental work, the paper systematically synthesizes and cross-analyses the supplied sources to construct a comprehensive set of architectural recommendations and operational controls. This design choice aligns with the task requirement to base findings strictly on the provided references and to elaborate theoretical implications and practical guidance with scholarly rigor.

Synthesis Procedure: The synthesis began by categorizing each reference by its primary contribution: technology comparison (AquaSec, 2019; Sahasrabudhe & Sonawani, 2014), platform deployment and orchestration guidance (Sehgal, 2012; Oracle, 2019), intrusion detection and prevention conceptualization and comparison (Ashoor & Gore, 2011; Singh & Singh, 2014), native multi-tenancy frameworks and enabling patterns (Guo et al., 2017; Almorsy et al., 2017; Flood & Keane, 2017), applied healthcare cloud considerations and privacy-preserving techniques (Marwan et al., 2017; Dharmaraju et al., 2016), and adversarial attack analyses relevant to cloud orchestration (Goutham & Tejaswini, 2016). Each category was then interrogated to extract core propositions, normative recommendations, and explicit technical details that could be transposed into an integrated framework.

Analytical Framing: The analysis employed layered architectural framing: physical and hypervisor layers, virtual machine and container runtimes, orchestration and management layers (OpenStack components), network fabric and software-defined networking, and application/multi-tenancy constructs. For each layer, the framework identifies primary assets, trust boundaries, typical threat vectors, and candidate controls referenced in the literature. This layered framing is explicitly derived from the native multi-tenancy and multi-tenant support literature (Guo et al., 2017; Almorsy et al., 2017) and from intrusion detection system literature which recommends placement and role distinctions between host-based IDS/IPS and network-based IDS/IPS (Ashoor & Gore, 2011; Singh & Singh, 2014).

Comparative Evaluation: To compare containerization and virtualization modalities, the study employed a conceptual evaluation matrix built from the container vs VM comparative article and industry guidance (AquaSec, 2019). The matrix examines criteria such as isolation strength, attack surface, resource utilization, start-up times, orchestration maturity, cryptographic integrity mechanisms, and operational monitoring implications. Each criterion is discussed in depth, and where appropriate reconciled with multi-tenant concerns raised in the native multi-tenancy literature (Guo et al., 2017; Almorsy et al., 2017).

Operational Hardening and OpenStack: OpenStack deployment guides and multi-node configuration documentation (Sehgal, 2012; Oracle, 2019) form the empirical backbone for orchestration and platform hardening recommendations. The methodology reinterprets the installation, node segmentation, and role assignment guidance within a security threat modeling perspective, deriving concrete configuration and segmentation strategies mapped to the layered analysis.

Privacy and Collaboration in Healthcare Clouds: Given the specific references to healthcare use-cases (Marwan et al., 2017; Dharmaraju et al., 2016; Righi et al., 2017), an applied evaluation is provided for secure data sharing, incorporating secure multi-party computation (MPC) as an enabled technique to permit cross-institutional collaboration while preserving patient privacy. This section synthesizes theoretical claims from Marwan et al. (2017) with applied system requirements discussed in hospital-centric IoT and elastic management literature (Righi et al., 2017).

Threat Modeling and IDS/IPS Analysis: The methodology includes a detailed threat taxonomy informed by documented adversarial strategies (Goutham & Tejaswini, 2016) and active detection frameworks for multi-tenant vulnerabilities (Flood & Keane, 2017). The IDS/IPS comparative literature (Ashoor & Gore, 2011; Singh & Singh, 2014) is used to map detection capabilities (e.g., signature versus anomaly, host vs network) onto the taxonomy.

Limitations and Scope: The methodological approach intentionally excludes primary data collection and retains strict adherence to the given reference list. By design, the framework prioritizes conceptual synthesis, normative operational guidance, and theoretical elaboration of mechanisms described in the supplied documents. Where the literature is silent or underspecified, the study offers reasoned theoretical inferences and denotes them as interpretative extrapolations anchored to cited material.

RESULTS

This section presents the synthesized outcomes of the analysis: an architectural risk taxonomy, a decision guide for choosing containers versus VMs in multi-tenant clouds, an operational hardening checklist for OpenStack multi-node environments, an intrusion detection and prevention mapping, and a privacy-aware blueprint for healthcare cloud collaboration.

Architectural Risk Taxonomy for Multi-Tenant Clouds

The first result is a comprehensive risk taxonomy organized by architectural layer, capturing the principal threat vectors that arise in multi-tenant clouds as synthesized from the literature.

Physical and Hypervisor Layer: At the lowest layer, co-tenancy risks primarily manifest as side-channel leakage, hardware resource contention, and hypervisor escape opportunities. The native multi-tenancy literature underscores that the foundational trust boundary often sits at the hypervisor or container engine level; therefore, any compromise here yields catastrophic cross-tenant exposure (Guo et al., 2017; Almorsy et al., 2017). Vulnerabilities at this layer include speculative execution attacks and other hardware side channels that leak secrets across tenants.

Virtualization and Container Runtimes: Containers reduce per-instance overhead but rely on shared kernel resources; their security model depends heavily on kernel capability separation and namespaces. Docker containers, as compared to full VMs, expose a smaller forward-facing management surface to the host but also depend on correct namespace and cgroup configuration to enforce isolation (AquaSec, 2019). Misconfiguration, insecure images, and elevated container capabilities are recurring sources of compromise.

Orchestration and Management Layer: Orchestration platforms (OpenStack, Kubernetes, etc.) are central trust concentrators: they manage identity, lifecycle, networking, and storage orchestration. Attack vectors include API abuse, credential compromise, role misassignment, and supply chain risks tied to third-party plug-ins. OpenStack multi-node deployment complexity increases the chance of inconsistent configuration and wide-blast radius misconfigurations (Sehgal, 2012; Oracle, 2019).

Network and SDN Fabric: Multi-tenant network overlays can be misconfigured, leading to VLAN hopping, route leakage, and insecure tenant traffic lateral movement. Network segmentation must be engineered carefully to maintain isolation while allowing required multi-tenant services (Guo et al., 2017).

Application and Data Layer: Logical multi-tenant design must avoid shared mutable state and ensure tenant separation in application logic. Native multi-tenancy patterns (tenant as schema, tenant as shared table, tenant as separate instance) each carry nuanced security consequences and must be chosen based on threat models and trust assumptions (Guo et al., 2017).

Human and Operational Layer: Threats include improper administrative practices, insufficient patching, and inadequate vulnerability scanning—particularly dangerous in multi-tenant contexts given privileged role proliferation and automation scripts that can make widespread changes (Almorsy et al., 2017; Flood & Keane, 2017).

Comparative Decision Guide: Containers vs Virtual Machines

The second result is a detailed comparative decision guide that explicates the trade-offs between Docker containers and traditional virtual machines in multi-tenant settings.

Isolation Strength: VMs provide kernel and hardware isolation by running separate operating systems on hypervisors; they therefore reduce shared kernel attack surface and are historically considered stronger for untrusted multi-tenant isolation (AquaSec, 2019). Containers share the host kernel, so isolation relies on namespaces, seccomp, AppArmor/SELinux, and cgroups; misconfigurations or kernel vulnerabilities can lead to cross-container escapes.

Attack Surface and Image Management: Containers enable rapid image distribution but increase attack surface if images are not curated. The literature emphasizes that container images should be treated as first-class security artifacts: scanned, signed, and provenance-tracked (AquaSec, 2019). By contrast, VMs are typically larger and slower to provision, but their image formats and update lifecycle may be simpler to integrate into traditional patch management.

Operational Agility and Density: Containers support higher density and faster scaling, which reduces cost and improves developer velocity. For workloads where tenants run similar code and share trust, containers provide substantial benefit. However, when tenants are untrusted or when regulatory requirements demand strong fault and privacy isolation (e.g., healthcare multi-tenant data), VMs may provide safer default isolation (Guo et al., 2017; Marwan et al., 2017).

Monitoring and Forensics: Containers' ephemeral nature complicates traditional forensic workflows; logs and forensic evidence must be externalized and preserved. VMs, being more persistent, facilitate snapshot-based forensics and system-wide memory capture. The IDS/IPS literature recommends host-level sensors that are container-aware and network sensors that understand overlay networks to maintain detection fidelity (Singh & Singh, 2014; Ashoor & Gore, 2011).

Cost-Security Tradeoffs: The decision is not binary; hybrid architectures often offer optimal compromise. For example, running tenant-facing microservices in containers atop VMs that provide kernel separation provides layered isolation (AquaSec, 2019; Guo et al., 2017).

OpenStack Multi-Node Operational Hardening Checklist

Synthesizing deployment guidance (Sehgal, 2012; Oracle, 2019) with multi-tenancy security research yields the following operational hardening checklist for OpenStack multi-node environments:

Role and Node Segmentation: Assign discrete roles to physical or virtual nodes (controller, compute, network, storage) and limit administrative access to role-specific operators; separate management network from tenant network plane (Sehgal, 2012; Oracle, 2019).

Least Privilege for APIs: Enforce least privilege for OpenStack service accounts and use scoped tokens. Auditable token lifetimes and robust logging are recommended.

Secure Configuration Management: Use immutable, versioned configuration repositories, automated configuration management tools, and policy-as-code to ensure consistent settings across nodes. Validate configurations post-deployment with active testing (Flood & Keane, 2017).

Image and Artifact Integrity: Maintain curated image catalogs, sign and verify images, and implement a vulnerability scanning pipeline for images prior to onboarding.

Network Segmentation and Overlay Controls: Implement tenant VLANs or virtual networks with robust enforcement at the hypervisor and software-defined networking (SDN) layers; use micro-segmentation where possible to enforce east-west traffic restrictions (Guo et al., 2017).

Monitoring, Logging, and Alerting: Centralize logs on secure, tamper-evident stores; ensure audit trails for administrative API calls; deploy IDS/IPS sensors that can interpret virtualized network contexts (Singh & Singh, 2014; Ashoor & Gore, 2011).

Patch and Vulnerability Management: Maintain an automated patching cadence for hypervisors, controller nodes, and container runtimes; test patches in staging to avoid orchestration disruptions.

Backup and Recovery: Plan for snapshot, backup, and rapid recovery for controller state and tenant data with strict access controls for restoration operations.

Active Vulnerability Scanning: Use frameworks for active detection of multi-tenant vulnerabilities and configuration drift as recommended by Flood & Keane (2017).

Intrusion Detection and Prevention Mapping

This result presents an operational mapping of IDS/IPS capabilities across architectural layers, drawing on Ashoor & Gore (2011) and Singh & Singh (2014).

Host-Based Intrusion Detection/Prevention (HIDS/HIPS): HIDS/HIPS are critical within tenant hosts (VMs or containers) to monitor file integrity, system calls, process anomalies, and local configuration changes. For containers, host-level agents must be container-aware; alternatively, lightweight agents can be embedded in container images but must be matched with secure image practices to prevent tampering (Ashoor & Gore, 2011).

Network-Based Intrusion Detection/Prevention (NIDS/NIPS): NIDS/NIPS observe traffic patterns and signatures on virtual or physical network taps. In multi-tenant overlays, instrumentation must consider encapsulation (VXLAN, GRE) and decrypt or inspect traffic where tenant privacy and legal frameworks permit. NIDS remains invaluable for detecting volumetric attacks and lateral movement.

Behavioral and Anomaly Detection: The literature supports anomaly detection techniques for identifying novel attacks and stealthy patterns that signature systems miss (Singh & Singh, 2014). Behavioral profiles should be built per tenant and per service to reduce false positives arising from diverse tenant behaviors.

Active Detection Frameworks: Flood & Keane (2017) advocate for frameworks that proactively test multi-tenant systems to expose misconfigurations and vulnerabilities. Active scanning combined with passive monitoring yields better visibility into the evolving security posture.

IPS Placement and Response: IPS should be placed where automated blocking will not disrupt tenant SLAs or violate isolation expectations. Configurations must balance rapid containment against risk of causing a denial of service through over-aggressive blocking.

Healthcare Cloud and Secure Multi-Party Computation (MPC)

Healthcare settings impose high privacy requirements on patient data, and the literature recommends integrating cryptographic and architectural techniques to enable secure collaboration (Marwan et al., 2017; Dharmaraju et al., 2016). The key result is a blueprint for privacy-preserving collaboration that combines MPC, secure orchestration controls, and strict access governance:

Data Minimization and Local Processing: Wherever possible, perform sensitive data processing within the data owner’s controlled environment and share only aggregated, minimized outputs.

Secure Multi-Party Computation: MPC enables parties to jointly compute functions over private inputs without revealing the inputs themselves, supporting collaborative analytics across hospitals without exposing raw patient records (Marwan et al., 2017). MPC can be integrated into cloud architectures as a specialized service or as a library deployed in tenant-controlled VMs to reduce trust in service providers.

Partitioned Data Architectures: Use logical partitions for identifiable data and pseudonymous or de-identified datasets for cross-tenant analysis to reduce risk surface (Righi et al., 2017).

Auditability and Consent: Maintain strict consent and audit trails for data access, supported by cryptographic proofs where feasible.

orchestration controls: Enforce strict role separation on orchestration platforms, limit sharing of images between institutions, and ensure cryptographic key management for data in motion and at rest.

Applied Attack Patterns: Denial-of-Service and Stealthy Orchestration Attacks

The literature documents advanced attack patterns that exploit orchestration and multi-tenant properties (Goutham & Tejaswini, 2016). The synthesized identification of these patterns and corresponding mitigations are:

Stealthy DDoS Orchestration: Attackers coordinate low-and-slow traffic across many tenant resources or exploit tenant APIs to trigger resource exhaustion on shared controllers. Mitigations include rate limiting, careful API quotas, and anomaly detection that accounts for distributed low intensity across tenants (Goutham & Tejaswini, 2016).

Credential Abuse and Lateral Movement: Compromised administrative credentials can grant broad control in multi-node deployments. Enforce multi-factor authentication, role-based access controls, and fine-grained audit logging to detect unusual administrative patterns (Sehgal, 2012; Oracle, 2019).

Supply Chain and Image Poisoning: Uncurated third-party images or plug-ins can introduce backdoors or covert channels. Maintain image signing, provenance checks, and vetted plugin repositories as defensive measures (AquaSec, 2019; Flood & Keane, 2017).

Active Detection Use Cases: The active detection frameworks suggested by Flood & Keane (2017) can be used to find lateral movement avenues and misconfigurations that enable stealthy attacks. Regular red-teaming exercises in the multi-tenant context are recommended.

DISCUSSION

This section interprets the results within theoretical and practical contexts, reflects on limitations, and illuminates directions for future research.

Theoretical Implications

Layered Defense and Trust Boundaries: The results reinforce a fundamental security principle: multi-tenant clouds require explicit, documented trust boundaries at each architectural layer (Guo et al., 2017). The hypervisor or container engine represents a pivotal trust anchor, and layered defenses—combining isolation, monitoring, and governance—are necessary because no single control offers complete protection (Almorsy et al., 2017).

Trade-off Theory between Agility and Isolation: The comparative analysis of containers and VMs exemplifies a recurring trade-off in secure systems design between agility/performance and strict isolation (AquaSec, 2019). The choice is not purely technical but depends on tenant trust relationships, regulatory environment, and sensitivity of data processed. The emergence of hybrid patterns (containers on dedicated VMs) reflects a pragmatic balancing of these competing desiderata.

Operationalization of Active Detection: Flood & Keane's (2017) advocacy for active detection reframes vulnerability management from a periodic scanning exercise to continuous, operations-integrated detection. The theoretical shift is toward systems that accept intrinsic misconfigurations and design for continuous discovery and rapid correction—reminiscent of resilience engineering paradigms.

The Role of Cryptography in Multi-Tenant Collaboration: The incorporation of MPC into cloud workflows for healthcare (Marwan et al., 2017) illustrates how cryptographic protocols can enable collaborative analytics without reliance on full trust in the cloud provider. Theoretically, this suggests a decomposition of trust into cryptographic assurances and operational assurances, enabling new service models where providers offer computation while cryptography preserves privacy.

Practical Implications and Recommendations

For Architects: Adopt hybrid deployment patterns where necessary; treat container images as security-sensitive artifacts; use VMs where tenant separation demands stronger kernel isolation (AquaSec, 2019; Guo et al., 2017).

For Operators: Implement the OpenStack hardening checklist—node segmentation, API privilege restrictions, centralized tamper-evident logging, and active vulnerability detection frameworks (Sehgal, 2012; Oracle, 2019; Flood & Keane, 2017).

For Security Practitioners: Deploy container-aware host sensors and overlay-aware network sensors; pair signature-based detection with anomaly models to capture stealthy orchestration attacks (Ashoor & Gore, 2011; Singh & Singh, 2014).

For Healthcare Institutions: Explore MPC for cross-institutional analytics; maintain strict key management and enforce least privilege across orchestration roles (Marwan et al., 2017; Dharmaraju et al., 2016).

LIMITATIONS

Reference Constraints: The analysis is constrained to the supplied references; this imposes limits on the currency and scope of the recommendations. For example, rapidly evolving technologies and specific vulnerabilities discovered after the dates of the referenced material are not covered.

Absence of Empirical Validation: The paper develops normative recommendations and conceptual mappings but does not provide empirical validation through measurements, controlled experiments, or case studies beyond what the cited works provided.

Generality versus Specificity: Multi-tenant systems vary widely; while the framework aims for general applicability, its operational recommendations must be adapted to particular platform versions, regulator regimes, and tenant profiles.

Future Research Agenda

Empirical Measurement Studies: Conduct longitudinal empirical studies comparing breach incidence and misconfiguration frequency across container-centric, VM-centric, and hybrid multi-tenant deployments.

Automated Misconfiguration Discovery: Develop and evaluate automated tools that detect inconsistent orchestration configurations in multi-node OpenStack deployments and recommend minimal corrective remediations.

MPC Performance and Usability: Investigate performance, scalability, and developer ergonomics of MPC libraries in production healthcare cloud workflows, including fault tolerance and latency-sensitive workloads.

Integrated IDS/IPS for Virtualized Overlays: Design intrusion detection systems that can natively parse and analyze overlay encapsulation technologies (VXLAN, Geneve) and that integrate tenant behavioral baselines to reduce false positives.

Policy and Governance Models: Explore policy frameworks that codify acceptable configurations and automate compliance enforcement across tenants without stifling tenant autonomy.

CONCLUSION

Securing multi-tenant cloud environments requires a synthesis of rigorous architectural reasoning, operational discipline, and advanced detection strategies. This paper, grounded in the supplied literature, presents an integrated framework that: defines a layered risk taxonomy; contrasts containers and virtual machines with an emphasis on isolation and operational implications; offers a detailed OpenStack multi-node hardening checklist; maps IDS/IPS functions to architectural layers; and proposes privacy-preserving collaboration strategies for healthcare clouds, notably leveraging secure multi-party computation. The synthesized recommendations advocate for layered defenses, provenance-aware image management, continuous detection, and cryptographic techniques to reduce trust dependencies. Practitioners should apply these recommendations conservatively and adapt them to context-specific constraints; researchers should pursue empirical validation and tool development to operationalize the theoretical insights. This integrated approach aims to help cloud architects and security teams navigate the complex trade-offs of multi-tenancy—achieving the efficiency and scaling benefits of shared infrastructure while preserving tenant confidentiality, integrity, and availability.

REFERENCES

1. AquaSec. Docker Containers vs. Virtual Machines. https://www.aquasec.com/wiki/display/containers/Docker+Containers+vs.+Virtual+Machines , Jul/2019.

2. Sehgal, Anuj. Introduction to OpenStack. Running a Cloud Computing Infrastructure with OpenStack, University of Luxembourg (2012).

3. Installing Across Multiple Systems for a Multi-node Havana OpenStack Configuration. https://docs.oracle.com/cd/E36784_01/html/E54155/installmulti.html#scrolltoc , Jul/2019.

4. Sahasrabudhe, Shalmali Suhas, and Shilpa S. Sonawani. ComparinOpenStackck aVMware. 2014 International Conference on Advances in Electronics Computers and Communications. IEEE,

2014.

5. Ashoor, Asmaa Shaker, and Sharad Gore. Difference between intrusion detection system (IDS) and intrusion prevention system (IPS). International Conference on Network Security and Applications. Springer, Berlin, Heidelberg, 2011.

6. Hariharan, R. Zero trust security in multi-tenant cloud environments. Journal of Information Systems Engineering and Management, 10. 2025.

7. Singh, Amrit Pal, and Manik Deep Singh. Analysis of host-based and network-based intrusion detection system. IJ Computer Network and Information Security 8 (2014): 41-47.

8. M. Marwan, A. Kartit, and H. Ouahmane. Applying secure multi-party computation to improve collaboration in healthcare cloud. Proc. - 2016 3rd Int. Conf. Syst. Collab. SysCo 2016, 2017.

9. C. J. Guo, W. Sun, Y. Huang, Z. H. Wang, and B. Gao. A Framework for Native Multi- Tenancy Application Development and Management A Native Multi-tenancy Enablement Framework Challenges of the Native Multi-tenancy Pattern. ECommerce Technol. 4th IEEE Int. Conf. Enterp. Comput. ECommerce Eser. 2017 CECEEE 2007 9th IEEE Int. Conf., pp. 551–558, 2017.

10. M. Almorsy, J. Grundy, and A. S. Ibrahim. SMURF: Supporting multi-tenancy using reaspects framework. Proc. - 2012 IEEE 17th Int. Conf. Eng. Complex Comput. Syst. ICECCS 2012, pp. 361–370, 2017.

11. J. Flood and A. Keane. A proposed framework for the active detection of security vulnerabilities in multi-tenancy cloud systems. Proc. - 3rd Int. Conf. Emerg. Intell. Data Web Technol. EIDWT 2012, pp. 231–235, 2017.

12. R. D. R. Righi, G. Rostirolla, C. A. Da Costa, M. Goulart, and E. Rocha. Elastic Management of Physical Spaces and Objects in Multi-Hospital Environments. Proc. - 2016 IEEE Int. Conf. Internet Things; IEEE Green Comput. Commun. IEEE Cyber, Phys. Soc. Comput. IEEE Smart Data, iThings-GreenCom-CPSCom-Smart Data 2016, pp. 33–38, 2017.

13. Gangu Dharmaraju, J. Divya Lalitha Sri and P. Satya Sruthi. A Cloud Computing Resolution in Medical Care Institutions for Patient’s Data Collection. International Journal of Computer Engineering and Technology, 7(6), 2016, pp. 83–90.

14. Dr. V. Goutham and M. Tejaswini. A Denial of Service Strategy To Orchestrate Stealthy Attack Patterns In Cloud Computing. International Journal of Computer Engineering and Technology, 7(3), 2016, pp. 179–186.