Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Analysis of Role-Based Access Control and Service Account Misconfigurations to Identify Vulnerabilities in Kubernetes

Ragini Modi[1],
Department of Computer Science
International Institute of Professional Studies (IIPS), DAVV
Indore, Madhya Pradesh, India

Nitin Nagar[2]
Department of Computer Science
International Institute of Professional Studies (IIPS), DAVV
Indore, Madhya Pradesh, India

Abstract - Kubernetes has emerged as widely used container orchestration platform for cloud-native applications, yet its complex configuration introduces significant security risks. Role-Based Access Control (RBAC) and Service Account (SA) address critical challenges for managing permissions within a cluster. This study presents a systematic analysis of RBAC and SA misconfigurations to identify vulnerabilities, their relationships, and corresponding countermeasures. The findings reveal that privilege escalation, unauthorized API access, secret exposure, and cluster compromise are among the most critical risks arising from these misconfigurations. The proposed classification provides a comprehensive reference for understanding Kubernetes security risks and establishes a foundation for future development of automated vulnerability detection and remediation mechanisms.

Keywords: ***Kubernetes Security, RBAC, SA Misconfiguration, Kubernetes Misconfiguration, Privilege Escalation, Container Orchestration Security.***

I. INTRODUCTION

Cloud computing has become a key platform for hosting and delivering modern applications. The increasing demand for scalable, flexible, and cost-effective computing resources has encouraged organizations to adopt cloud-native architectures [1]. Container technology has also gained widespread adoption due to its ability to package applications along with their dependencies and enable consistent deployment across different platforms. It offers a lightweight, portable, and efficient solution for large-scale application deployment [2]. The rapid growth of containerized applications has increased the complexity of container management. Managing large-scale container environments manually often leads to inefficiencies and operational challenges related to deployment, scaling, networking, and resource allocation [3]. These operational challenges have driven the adoption of container orchestration platforms [4].

Kubernetes is widely recognized as a leading container orchestration platform capable of managing complex containerized workloads efficiently. It provides services such as, automated scheduling, service discovery, load balancing, self-healing, and resource optimization, which simplify the deployment and management of cloud-native applications [4]. Despite its various features, Kubernetes environments face significant security challenges and misconfiguration issues due to their distributed nature and the complexity of managing multiple interconnected components [5][6]. Common security challenges include inadequate access control, excessive privileges, insecure default configurations, insufficient workload isolation, weak network segmentation, and unauthorized access to sensitive resources [7]. Kubernetes misconfiguration risks such as, improper or insecure configuration of security, networking, or access control parameters [8]. These misconfigurations lead to serious risks while accessing the Kubernetes resources. RBAC and SA serve as key mechanism for regulating access to resources. RBAC defines and enforces permissions assigned to users, groups, and services, whereas SA provide identities for applications and pods operating within the environment [10].

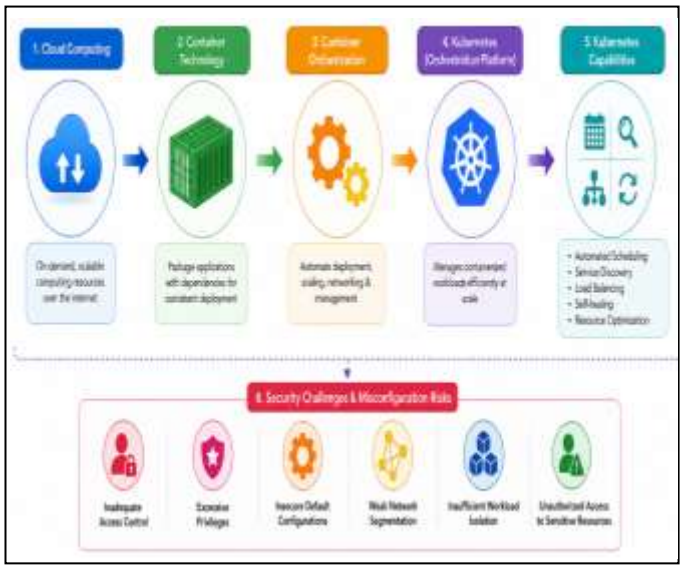

Figure 1: Cloud-Native Application Evolution and Kubernetes Management Framework

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Misconfigurations in RBAC policies and SA permissions can grant unnecessary privileges, creating opportunities for unauthorized access, privilege escalation, and the compromise of critical system resources [7]. This violates the principle of least privilege and increases the risk of privilege escalation along with unauthorized access to system resources [7][10]. SA misconfigurations occur when application identities are assigned excessive permissions. This leads to workloads gaining unintended access to sensitive resources and APIs, resulting in data exposure or potential system compromise [11][12]. Misconfigurations in both RBAC and SA can introduce security vulnerabilities such as, privilege escalation, unauthorized resource access, secret exposure, Kubernetes API abuse, and cluster compromise, which further increase the risk of unauthorized operations, data breaches, service disruptions, and compromise of critical cluster resources [7][11]. Therefore, this study analyzes RBAC and SA misconfigurations to identify key vulnerabilities and threats that impact Kubernetes security. This classification helps in understanding the relationship among misconfigurations, vulnerabilities, threats, and their countermeasures.

The paper is organized as follows: Section I introduces the background, motivation, and objectives of the study. **Section II** presents the literature review on Kubernetes security, RBAC, and SA misconfigurations. **Section III** presents the analysis of RBAC and SA misconfigurations, associated vulnerabilities, threats, their corresponding countermeasures. **Section IV** discusses the findings and outlines potential directions for future work. **Section V** concludes the paper by summarizing the key findings and highlighting future research directions.

II. LITERATURE REVIEW

The existing research on Kubernetes security focuses on access control mechanisms and misconfiguration issues. It highlights key studies on RBAC and SA and their misconfigurations contribute to security vulnerabilities and associated threats in Kubernetes environments.

A. Kubernetes security challenges

Shamim *et al.,* introduced the "XI Commandments of Kubernetes Security," which outlined essential security practices covering authentication, authorization, network security, workload protection, and secret management. The study emphasized that weak access control and insecure configurations are among the primary sources of security risks in Kubernetes environments [7]. Chittibala, reviewed multiple Kubernetes security practices and discussed the necessity of securing cluster resources, workloads, network communication, and access control mechanisms. The study focuses on the importance of implementing security controls to mitigate risks associate to unauthorized access and misconfigurations in Kubernetes [6].

T. Ghadiya and H. Trivedi, examined security threats in Kubernetes and discussed various best practices along with hardening techniques for securing Kubernetes deployments. The work highlighted how security controls play a crucial role in reducing threats and enhancing the overall security of Kubernetes clusters [13]. Kampa, analyzed the security threats and challenges in Kubernetes environments and highlighted the importance of securing cluster resources. The study discussed the role of security measures in reducing security risks and improving Kubernetes security [15]. Collectively, these studies show that securing Kubernetes requires a combination of robust access control policies, workload isolation, continuous monitoring, and effective configuration management.

B. Kubernetes misconfiguration challenges

Zhang *et al.,* studied configuration defects in Kubernetes by analyzing 719 defects from 2,260 configuration files. They identified 15 categories of misconfiguration issues and showed that such defects are common and can lead to serious system failures. The study demonstrated that many vulnerabilities originate from configuration weaknesses rather than flaws in Kubernetes itself [8]. Russell and Dev, proposed a centralized defense approach for Kubernetes that focuses on improving security through effective logging and mitigation strategies. The work highlighted the importance of continuous auditing and monitoring for detecting insecure configurations before they can be exploited. The authors emphasized that automated detection mechanisms are essential for maintaining security in large-scale Kubernetes deployments [17].

Malul *et al.,* proposed GenKubeSec, an LLM-based framework for detecting, localizing, reasoning about, and remediating Kubernetes misconfigurations. The study shows how large language models can improve the identification of configuration issues and provide automated guidance for fixing them. The study highlights that integrating LLM-based approaches can enhance misconfiguration detection and support more effective remediation in Kubernetes environments [12].

C. RBAC security challenges

Rostami, examined RBAC authorization in Kubernetes and highlighted its role in managing access to cluster resources. The study explains that RBAC provides fine-grained permission control, improper role definitions and excessive privileges can lead to security risks. It further emphasizes that misconfigurations in RBAC policies may result in unauthorized access and weaken the overall security posture of Kubernetes environments [10]. Amgothu and Kankanala, examined methods for enhancing Kubernetes security through workload protection and RBAC optimization. The research highlighted that excessive permissions assigned to users, groups, and SA frequently violate the principle of least

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

privilege. Such configurations increase the unauthorized access and privilege escalation attacks. The authors recommended periodic permission reviews, role minimization, and continuous RBAC auditing as effective mitigation strategies [9]. Additionally, industry threat reports indicate that attackers often exploit misconfigured RBAC policies and service account tokens to gain unauthorized access and escalate privileges within Kubernetes clusters [18]. Overall, existing studies show that while RBAC is a fundamental security mechanism in Kubernetes, misconfigurations and overly permissive policies remain a persistent challenge, highlighting the need for more robust and automated RBAC approach.

D. SA security challenges

Souppaya *et al.*, emphasize that improper handling of container and application identities, along with excessive privilege allocation, can degrade overall security posture and expose systems to potential misuse of credentials. The work highlights the importance of strict access control and secure identity management in container environments to reduce attack opportunities [2]. Yang *et al*, highlight that container-based cloud environments face several security challenges due to weak identity management and improper handling of these service credentials. The study shows that when service accounts are configured with excessive permissions or lack proper isolation, they can become a potential entry point for attackers to gain unauthorized access to cluster resources [11]. Gajbhiye *et al.,* emphasize that poor vulnerability management and weak access governance in Kubernetes environments contribute to security exposure, especially when service accounts are not properly isolated or regularly reviewed. These issues can be exploited to escalate privileges and access restricted resources within the cluster [16]. Although Kubernetes security has been widely studied, most existing work focuses on either general security issues or isolated discussions of RBAC and SA configurations. As a result, there is limited work that collectively analyzes how these misconfigurations lead to specific vulnerabilities, associated threats, and possible countermeasures in a unified manner. The lack of integrated analysis motivates the present study, which systematically classifies RBAC and SA misconfigurations along with their associated vulnerabilities, threats, and mitigation strategies.

III. VULNERABILITIES, THREATS, AND THEIR COUNTERMEASURES

The identified vulnerabilities, threats, and countermeasures related to RBAC and SA misconfigurations are organized in the framework shown in Figure 2. The framework presents the relationship between RBAC and SA misconfigurations, associated vulnerabilities, threats, and corresponding countermeasures. It provides overview of security risks that can arise from RBAC and SA misconfigurations in Kubernetes [14].

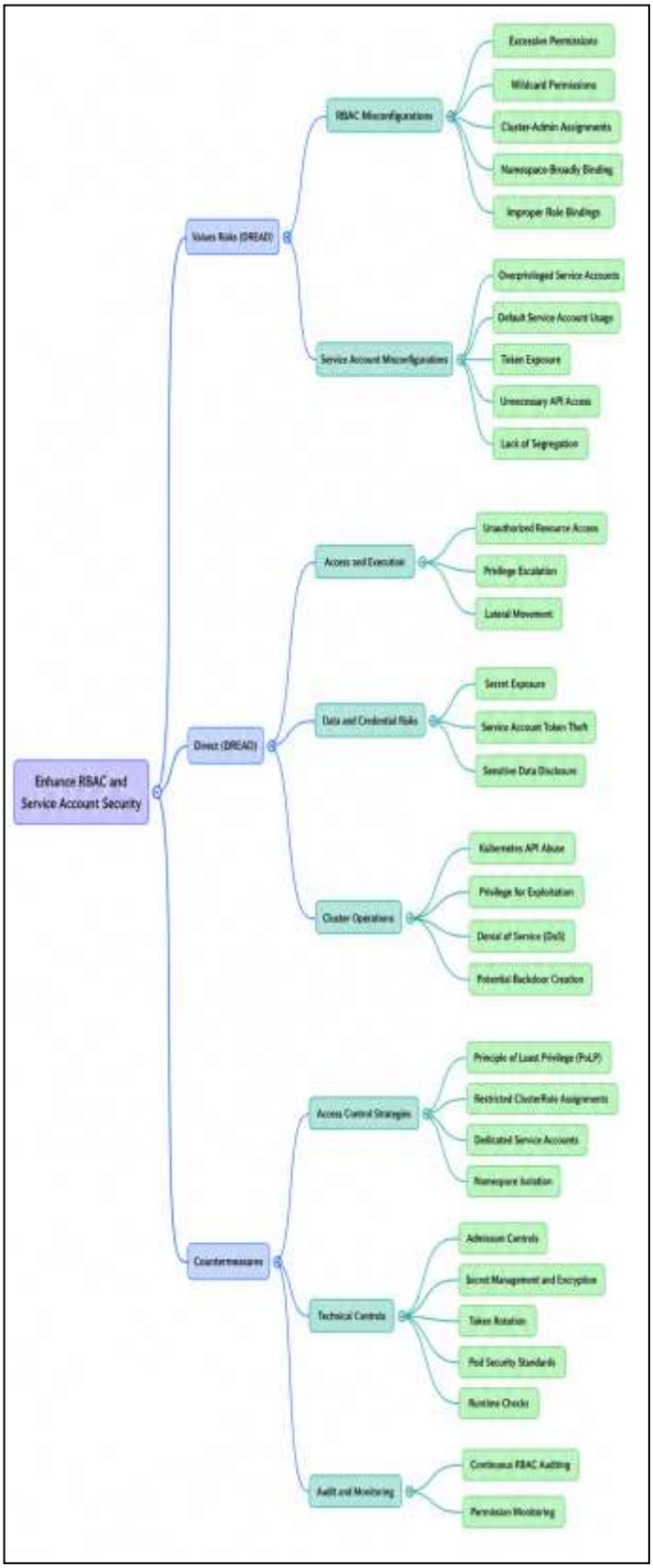

Figure 2: RBAC and SA Security Framework

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Table 1. Kubernetes Vulnerabilities (KVID) in RBAC and SA Misconfiguration

Vulnerability ID	Vulnerability	Description	Kubernetes Component
$KVID_1$	Excessive RBAC Permissions [7], [9]	• Roles grant more permissions than required. • Violates least privilege principle. • Enables unauthorized access.	RBAC
$KVID_2$	Wildcard Permissions [8]	• Uses "*" in verbs/resources. • Grants unrestricted access. • Easily exploitable.	RBAC
$KVID_3$	Cluster-Admin Assignment [7][10]	• Full cluster control assigned. • Allows critical changes. • High privilege escalation risk.	RBAC
$KVID_4$	Namespace Boundary Violation [5][7]	• Cluster-wide access instead of namespace. • Breaks isolation rules. • Increases lateral movement risk.	RBAC
$KVID_5$	Improper Role Bindings [10]	• Incorrect assignment of users, or groups. • Unauthorized user access. • Weak access control.	RBAC
$KVID_6$	Overprivileged SA [11][12]	• Extra permissions to workloads. • Unnecessary API access • High security exposure.	SA
$KVID_7$	Default SA Usage [12]	• Uses default SA without restriction. • Broad implicit permissions. • Common misconfiguration.	SA
$KVID_8$	SA Token Exposure [2][12]	• Tokens exposed through insecure storage or mounting. • Enables credential theft and API impersonation. • Increases unauthorized access risk.	SA
$KVID_9$	Unrestricted API Access [11]	• Direct API server access. • No permission filtering. • High abuse potential.	SA
$KVID_{10}$	Lack of SA Segregation [12]	• Shared SA across apps. • No isolation between workloads. • Easier attack propagation.	SA

Table 2. Kubernetes Threats (KTID) in RBAC and SA Misconfiguration

Threat ID	Threat	Description	Impact
$KTID_1$	Unauthorized Resource Access [5][7]	• Access beyond allowed permissions. • Exploits RBAC flaws. • Reads sensitive data.	Confidentiality Breach.
$KTID_2$	Privilege Escalation [7][11]	• Low privilege becomes admin. • Exploits misconfigured roles. • Gains full control.	Full Cluster Compromise.
$KTID_3$	Kubernetes API Abuse [11]	• Misuse of API server. • Unauthorized operations. • Cluster modification.	Integrity Violation.
$KTID_4$	Secret Exposure [2][12]	• Access to Kubernetes Secrets. • Credential theft. • Sensitive data leakage.	Data Leakage.
$KTID_5$	Lateral Movement [5][11]	• Access across namespaces. • Exploits weak isolation. • Spreads compromise.	Expanded Attack Surface.
$KTID_6$	SA Token Theft [12]	• Stealing SA tokens. • Impersonation of workloads. • Unauthorized API calls.	Unauthorized Access.
$KTID_7$	Privileged Pod Deployment [7][11]	• Deployment of high privilege pods. • Executes malicious workloads. • Gains node access.	Cluster Takeover.
$KTID_8$	Denial of Service (DoS) [5]	• Resource deletion or overload. • Cluster disruption. • Service downtime.	Service Disruption.
$KTID_9$	Persistent Backdoor Creation [11]	• Creates hidden access points. • Maintains attacker access. • Long-term compromise.	Long-Term Persistence.
$KTID_{10}$	Sensitive Data Disclosure [2]	• Configuration and Secret Exposure. • Internal data leakage. • Confidentiality loss.	Information Exposure.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

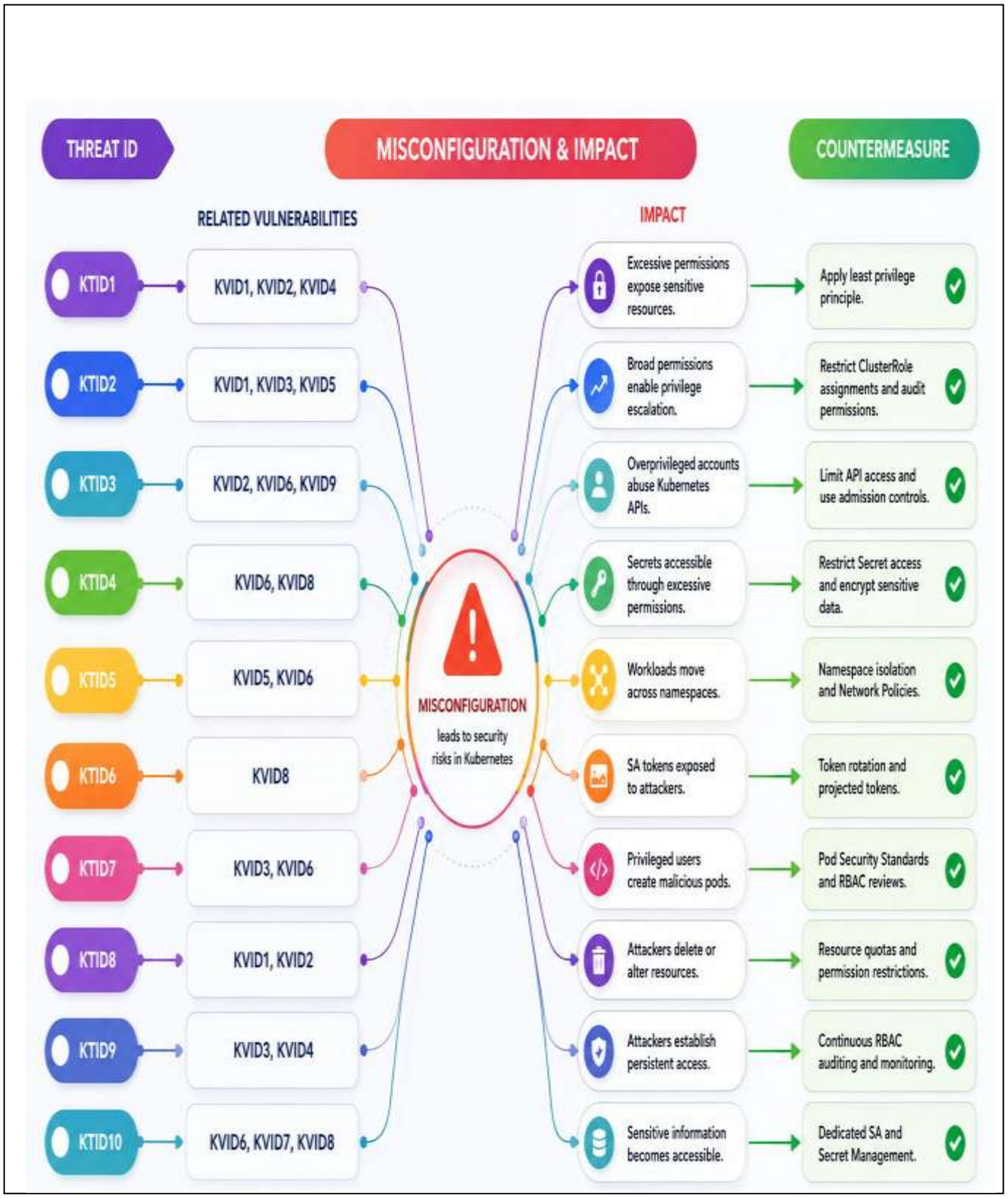

Figure 3: Mapping of Kubernetes Threats, Vulnerabilities, and Corresponding Countermeasures

The figure 3 presents the relationship between identified Kubernetes threats, associated vulnerabilities, and their recommended countermeasures. Each threat is mapped to one or more RBAC and SA misconfiguration vulnerabilities, highlighting potential security risks within the cluster. The corresponding countermeasures provide mitigation strategies to reduce the attack surface, prevent privilege escalation, and strengthen overall Kubernetes security.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Figure 4: Security Improvements Achieved Through Kubernetes Countermeasures and Mitigation Strategies

The figure 4. summarizes the countermeasures proposed to address the identified Kubernetes RBAC and SA vulnerabilities. These measures are intended to mitigate potential security threats, enforce proper access control, and enhance the overall security posture of Kubernetes environments

IV. Discussion and Future Work

This study presents a structured analysis of RBAC and SA misconfigurations in Kubernetes and explains their relationship with security concerns. The finding shows that misconfigurations such excessive permissions, using wildcard permissions, incorrect role assignments, and insecure SA settings can reduce the security of Kubernetes clusters. There is a clear relationship between misconfigurations, vulnerabilities, and threats. Most security issues occur when the least privilege principle is not followed. In such cases, users or services receive more permissions than required, which can lead to problems like privilege escalation, unauthorized access, and secret exposure.

The analysis also indicates that SA misconfigurations can be more critical because they directly affect running applications and may result in full cluster compromise if misused. RBAC misconfigurations such as, cluster-admin access and wildcard permissions can also increase risk by allowing access to more resources than needed. Overall, the study shows that Kubernetes security mainly depends on correct configuration rather than system weaknesses. Therefore, proper permission management, regular checking of access rights, and following the least privilege principle are important for securing Kubernetes environments.

V. Conclusion

This study presents a systematic analysis of RBAC and SA misconfigurations in Kubernetes and their security impact. A clear structure is created to group different vulnerabilities and threats, illustrating how misconfigurations lead to security issues and their countermeasures. The analysis shows that excessive permissions, wildcard role assignments, insecure SA usage, and improper token handling are the major security risks in Kubernetes environments. These issues can lead to serious problems such as, privilege escalation, unauthorized access, secret exposure, movement across the cluster, and even full cluster compromise. The proposed classification helps in understanding access control-related security risks in Kubernetes and can be useful for improving misconfiguration management practices. In future work, automated tools can be developed to detect and remediate these misconfigurations, along with testing the proposed framework in real Kubernetes environments.

References

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology, Gaithersburg, MD, Special Publication 800-145, Sept. 2011.

[2] M. Souppaya, J. Morello, and K. Scarfone, "Application Container Security Guide," National Institute of Standards and Technology, Gaithersburg, MD, Special Publication 800-190, Sept. 2017.

[3] S. Sultan, I. Ahmad, and T. Dimitriou, "Container Security: Issues, Challenges, and the Road Ahead," *IEEE Access*, vol. 7, pp. 52976-52996, Apr. 2019.

[4] Kubernetes Authors, "Kubernetes Components," Kubernetes Documentation. [Online]. Available: https://kubernetes.io/docs/concepts/overview/components/. Accessed: May 31, 2026

[5] Y. Yang, W. Shen, B. Ruan, W. Liu, and K. Ren, "Security challenges in the container cloud," *Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications*, pp. 137–145, Dec. 2021, doi: 10.1109/tpsisa52974.2021.00016.

[6] D. R. Chittibala, "Security in Kubernetes: A Comprehensive Review of best Practices," *International*

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Journal of Science and Research (IJSR), vol. 12, no. 6, pp. 2966–2970, Jun. 2023, doi: 10.21275/sr24304111526

[7] M. S. I. Shamim, F. A. Bhuiyan, and A. Rahman, "XI Commandments of Kubernetes Security: A Systematization of Knowledge Related to Kubernetes Security Practices," in *2020 IEEE Secure Development (SecDev)*, Atlanta, GA, USA, 2020, pp. 58-64.

[8] Y. Zhang, U. Paul, M. d'Amorim and A. Rahman, "Configuration Defects in Kubernetes," in IEEE Transactions on Software Engineering, vol. 52, no. 3, pp. 1125-1144, March 2026, doi: 10.1109/TSE.2026.3659785.

[9] S. Amgothu and G. Kankanala, "Enhancing Kubernetes Security: Securing Workloads and Optimizing Role-based Access Control," *International Journal of Computer Applications*, vol. 186, no. 58, pp. 11-15, Dec. 2024.

[10] G. Rostami, "Role-based Access Control (RBAC) Authorization in Kubernetes," *Journal of ICT Standardization*, vol. 11, no. 3, pp. 237–260, 2023.

[11] N. Yang, W. Shen, J. Li, X. Liu, X. Guo, and J. Ma, "Take Over the Whole Cluster: Attacking Kubernetes via Excessive Permissions of Third-party Applications," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23)*, Copenhagen, Denmark, 2023, pp. 3048–3062.

[12] E. Malul, Y. Meidan, D. Mimran, Y. Elovici, and A. Shabtai, "GenKubeSec: LLM-Based Kubernetes Misconfiguration Detection, Localization, Reasoning, and Remediation," *arXiv preprint arXiv:2405.19954*, May 2024, doi: 10.48550/arXiv.2405.19954.

[13] T. Ghadiya and H. Trivedi, "Kubernetes Security: A Review of Threats, Best Practices, and Real World Hardening Techniques," *International Journal of Innovative Research in Technology (IJIRT)*, vol. 12, no. 11, pp. 4327–4334, Apr. 2026.

[14] N. Nagar and U. Suman, "Analyzing Virtualization Vulnerabilities and Design a Secure Cloud Environment to Prevent from XSS Attack," *International Journal of Cloud Applications and Computing (IJCAC)*, vol. 6, no. 1, pp. 1-14, Jan.-Mar. 2016.

[15] S. Kampa, "Navigating the Landscape of Kubernetes Security Threats and Challenges," *Journal of Knowledge Learning and Science Technology*, vol. 3, no. 4, pp. 274–281, 2024.

[16] B. Gajbhiye, O. Goel, and P. K. G. Pandian, "Managing Vulnerabilities in Containerized and Kubernetes Environments," *Journal of Quantum Science and Technology*, vol. 1, no. 2, pp. 59-71, 2024.

[17] E. Russell and K. Dev, "Centralized Defense: Logging and Mitigation of Kubernetes Misconfigurations with Open Source Tools," *arXiv preprint arXiv:2408.03714*, Aug. 2024, doi: 10.2139/ssrn.4869634.

[18] Unit 42 (Palo Alto Networks), "Modern Kubernetes Threats," 2024. [Online]. Available: https://unit42.paloaltonetworks.com/modern-kubernetes-threats/ . [Accessed: May 31, 2026].