frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 07 July 2025
DOI 10.3389/fcomp.2025.1557918

OPEN ACCESS

EDITED BY
Silvio Ranise,
University of Trento, Italy

REVIEWED BY
Mengmeng Ren,
Xidian University, China
Tarun Kumar Vashishth,
IIMT University, India

*CORRESPONDENCE
Mohammad Hafiz Hersyah
✉ mohammad.hafiz_hersyah.mc4@is.naist.jp

RECEIVED 09 January 2025
ACCEPTED 16 June 2025
PUBLISHED 07 July 2025

CITATION
Hafiz Hersyah M, Hossain MD, Taenaka Y and Kadobayashi Y (2025) Fuzzyfortify: a multi-attribute risk assessment for multi-factor authentication and cloud container orchestration.
Front. Comput. Sci. 7:1557918.
doi: 10.3389/fcomp.2025.1557918

Fuzzyfortify: a multi-attribute risk assessment for multi-factor authentication and cloud container orchestration

Mohammad Hafiz Hersyah[1,2]*, Md. Delwar Hossain[3], Yuzo Taenaka[1] and Youki Kadobayashi[1]

[1]Cyber Resilience Laboratory, Division of Information Science, Nara Institute of Science and Technology, Ikoma - Nara, Japan, [2]Computer and Networking Laboratory, Information Technology Faculty, Andalas University, Padang, Indonesia, [3]Department of Computer Science, Angelo State University, San Angelo, TX, United States

Securing cloud-native infrastructures that integrate Multi-Factor Authentication (MFA) via FIDO2, container orchestration with Kubernetes, and Dockerized microservices remains a complex challenge due to interdependent vulnerabilities and escalating adversarial threats. To address this, we propose a web-based cybersecurity framework that combines Fuzzy Analytical Hierarchy Process (Fuzzy AHP), Domain Mapping Matrix (DMM), and fuzzy inference to perform multi-attribute risk assessment tailored to containerized environments. The method involves aggregating expert judgments to prioritize six key CIA-AAN criteria-Confidentiality, Integrity, Availability, Authentication, Authorization, and Non-repudiation-followed by structural complexity quantification using DMM enhanced with Singular Value Decomposition. These are then fused into a Complexity Resilience Index and used in a fuzzy logic system that incorporates CVE-derived indicators such as base score, impact, and exploitability. When applied to five real-world adversarial techniques, the framework produced differentiated risk outcomes: Data Destruction and Resource Hijacking emerged as High-Level Risks with scores of 70.47 and 74.60 respectively, while Endpoint DOS, Network DOS, and Inhibit System Recovery were classified as Medium-Level Risks. These results illustrate how layered threat propagation and component interdependence increase vulnerability in FIDO2-integrated orchestration settings. Compared to conventional frameworks like EBIOS and NIST RMF, our approach offers enhanced granularity in quantifying risk and simulating threat propagation. By enabling practitioners to understand not only which adversarial activities are most damaging but also why, this framework empowers more informed and proactive cybersecurity decisions-bridging the gap between technical risk modeling and real-world defense planning.

KEYWORDS

MFA, Docker, Kubernetes, fuzzy logic, multi-attribute risk assessment, cloud computing

1 Introduction

Cloud computing has transformed how services are developed, deployed, and managed. It enables automation, scalability, and continuous delivery pipelines, allowing organizations to respond quickly to business and user needs (Jun, 2017). As a result, the cloud-native market, valued at USD 794.1 million in 2021, is projected to reach USD 9,621.39 million by 2031 (Business Insight, 2023). A core enabler of this growth is the cloud-native paradigm, which emphasizes modularity and portability.

Application components are independently packaged and deployed across heterogeneous infrastructures, promoting agility and efficient resource utilization. Containerization plays a central role in supporting this modular architecture by delivering lightweight, portable services. At scale, container orchestration platforms such as Docker Swarm, Kubernetes, and Apache Mesos automate deployment, replication, failover, and system scaling (Lee et al., 2021). Security in distributed and dynamic environments depends heavily on reliable access control mechanisms. Multi-Factor Authentication (MFA) has emerged as an essential layer of defense for cloud-native systems. Among existing standards, FIDO2 has gained adoption as a passwordless authentication protocol that eliminates shared secrets and mitigates phishing risks (FIDO, 2022; Ghorbani Lyastani et al., 2020). It supports secure authentication across web and non-web services. However, integrating FIDO2 with orchestration systems such as Kubernetes introduces complex vulnerabilities. Internal misconfigurations and software inconsistencies, combined with the growing reliance on third-party services, increase the likelihood of exposure to multi-dimensional threats (Grimes, 2020).

This study introduces a prioritization strategy for security properties through the modified fuzzy analytical hierarchy process (fuzzy AHP). The method handles uncertainty by aggregating expert judgment and assigning fuzzy weights to the CIA-AAN criteria: Confidentiality, Integrity, Availability, Authentication, Authorization, and Non-repudiation (Bhol et al., 2023; Taleby Ahvanooey et al., 2023; Ogundoyin and Kamil, 2020). These properties require a unified perspective. Isolated emphasis on usability or confidentiality alone can result in system vulnerabilities. Authentication establishes identity (Kim et al., 2020), authorization defines access through RBAC (Zahoor et al., 2023), and non-repudiation ensures user accountability (Schiavone et al., 2016).

The architectural security dimension is captured using the Domain Mapping Matrix (DMM), which models interdependencies among components, interfaces, and system layers. This matrix, combined with Singular Value Decomposition (SVD), allows structural complexity in FIDO2-integrated environments to be measured quantitatively (Sinha et al., 2014). Results from this complexity analysis are then integrated with the security priority weights from modified Fuzzy AHP, forming a novel metric: the Complexity Resilience Index.

Risk levels are assessed using fuzzy logic, which incorporates the complexity resilience index, and real-world CVE data such as impact, base score, and exploitability. Fuzzy inference rules support evaluation under uncertain or incomplete data conditions, offering dynamic and context-aware risk assessments (Outkin et al., 2023; Blaise and Rebecchi, 2022; Gao et al., 2019).

Traditional risk assessment methods often fail to capture such complexity. For example, EBIOS Risk Manager (de la Sécurité des Systémes d'Information, 2019) provides structured analysis but focuses primarily on external threats, lacking integration with architectural dependencies or empirical threat intelligence. Moreover, frameworks like EBIOS do not define thresholds for acceptable risk and rely heavily on generalized mitigation actions. Similar limitations exist in recent literature (Wong et al., 2023; Sultan et al., 2019), where qualitative mitigation evaluations are rarely linked to actionable, data-driven insights.

The proposed framework addresses these gaps by combining expert-based decision-making, structural complexity modeling, and threat intelligence within a unified, multi-attribute risk assessment model. It supports FIDO2, Kubernetes, and Docker environments and is implemented as a web-based simulation tool. Practitioners can use this platform to explore attack scenarios, assess evolving risks, and test mitigation strategies in real time. By aligning theoretical models with operational demands, the framework supports more informed and adaptive cybersecurity decisions.

1.1 Motivation

Risk assessment in cloud-native security remains an unresolved challenge, particularly when dealing with technologies like FIDO2-based MFA, Kubernetes, and Docker. Existing frameworks often rely on general scoring models, static weights, or loosely structured matrices to evaluate critical system properties such as confidentiality, integrity, and availability. These methods struggle to produce consistent prioritization, especially when expert judgment varies or when evaluations involve user-centered concepts like authentication, authorization, and non-repudiation. Without a clear structure, organizations struggle to set effective security priorities.

Beyond this, current approaches such as de la Sécurité des Systémes d'Information (2019) fail to account for the structural complexity of modern architectures. Asset relationships, interface exposures, and dependency layers are rarely quantified in existing assessments, even though these structural elements often influence how vulnerabilities manifest in real systems. Despite the growing adoption of DevOps and orchestration tools in production, there remains a disconnect between best practice guidelines and how architectural complexity impacts risk exposure and the effectiveness of implemented controls.

A similar issue exists in how mitigation strategies are planned and executed. Security controls are often applied as qualitative, and guided by checklist-based practices (Wong et al., 2023; Sultan et al., 2019). There is rarely a structured method to assess how layered defenses, such as detection, mitigation, and prevention, work in combination to reduce residual risk. In time-sensitive or resource-constrained environments, this often results in inconsistent decisions and suboptimal allocation of security investments. What remains lacking is a systematic and quantifiable approach to align threat exposure, system architecture, and control effectiveness, so that mitigation planning becomes both defensible and operationally effective.

1.2 Novelty

This paper introduces a novel, multi-attribute risk assessment framework for cloud-native security. It integrates:

- **Modified fuzzy AHP**: We incorporate basic AHP into pairwise comparison by aggregating the mean of all experts' central consensus judgments using the Saaty scale, followed by fuzzification to compute the normalized value. Existing Fuzzy

AHP, like (Chang, 1996), computes fuzzy synthetic extents per expert, then compares fuzzy values using the degree of possibility.

- **Domain mapping matrix**: We correlate industry best practices and asset provision of Kubernetes, Docker, and FIDO2 for MFA in a cloud computing environment using singular value decomposition to determine structural complexity metrics: components, interfaces, and architecture. To our knowledge, no cybersecurity studies have modeled structural complexity using Domain Mapping Matrix or SVD, despite its use in systems engineering (Sheard and Mostashari, 2009).
- **CISSP-based risk reduction with SAFe scaling**: This study proposes a risk reduction strategy from the CISSP principle of layered defense—detection, mitigation, and prevention—each contributing incrementally to lowering residual risk (Chapple et al., 2018). To operationalize this concept, the approach adopts the Scaled Agile Framework's (SAFe) Weighted Shortest Job First (WSJF) model (Knaster and Leffingwell, 2020), applying tiered effectiveness weights of 5%, 3%, and 1% to reflect the cumulative impact of layered controls in prioritizing cybersecurity mitigation efforts.

1.3 Contribution

This study provides three primary contributions:

1. **Complexity resilience index**: we formulate a novel index that fuses structural complexity metrics from domain mapping (components, interfaces, architecture) with modified Fuzzy AHP-based CIA-AAN prioritization to quantify system resilience.
2. **Fuzzy logic-based risk assessment using real-world threat intelligence**: we apply fuzzy logic to combine the complexity resilience index with CVE-based threat metrics (impact, base score, exploitability) for five adversarial techniques—data destruction, endpoint denial of service, network denial of service, inhibit system recovery, and resource hijacking.
3. **Web-based implementation**: we deploy the core framework as an interactive, web-based tool to facilitate practitioners' adoption. The platform allows users to conduct what-if simulations and visualize changes in risk levels based on varying structural and threat inputs.

We divide this paper into seven (7) sections. Section 2 provides related prior research. Section 3 explains the assets domain and adversarial techniques. Section 4 discusses methodology. Section 5 presents the proposed multi-attribute risk assessment activities in detail. Section 6 discusses the comparative analysis of risk evaluation techniques, the impacts of adversarial methods, and the framework's limitations. The paper concludes with future research directions in Section 7.

2 Related prior research

This section presents prior research on multi-factor authentication (MFA) security, container orchestration security, fuzzy logic, risk assessment-based methodology, and mitigation strategies.

2.1 Multi-factor authentication security

A paper from Derhab et al. (2020) studies the security of the proposed architecture. It also evaluates a two-factor mutual authentication protocol for mobile cloud computing. Using MFA can spot early signs of compromise. It can find hacked accounts using advanced logs. Logs show that users who authenticate may decline or time out during the second phase of the method, per (Henricks and Kettani, 2020). This can trigger specific security rules and brief the analyst on the incident. According to Pöhn et al. (2023), security flaws are not always in MFA mechanisms themselves. This highlights social engineering as a critical next concern. An adversary could take advantage of this to conduct malicious activities.

2.2 Container orchestration security

The deployment of Kubernetes in large-scale systems like Netflix and Uber demonstrates its ability to manage extensive container ecosystems. It also highlights security vulnerabilities requiring thorough risk assessments (Nguyen, 2023). While our previous work (Hersyah et al., 2023) proposed a multi-dimensional risk assessment for Docker containers in IaaS environments using tools like AHP, ISO 27K, and MITRE (Adversarial Tactics, Techniques, and Common Knowledge) ATT&CK, it had limitations, including on Docker assets, inadequate guidance on resource limits, and a lack of real-world Kubernetes attack scenarios. Papers Mostajeran et al. (2017) and Blaise and Rebecchi (2022) further explore containerized platform risks and Helm Chart deployments, identifying vulnerabilities but lacking systematic methodologies and comprehensive Kubernetes threats. Additionally, Minna et al. (2021); Cao et al. (2022) examine Kubernetes networking and security abstractions but fall short in detailed risk profiling. Building on these findings, our study aims to develop a holistic multi-attribute risk assessment framework.

2.3 Fuzzy logic

The work in Flavia and Chelliah (2023) proposed an optimized, fuzzy logic-based method. It aims to create an anonymous identity and authenticate users. This would allow them to exchange data securely within P2P cloud environments. By addressing CIA-AAN, we seek a holistic approach. We will consider multiple facets of the foundation of cybersecurity. Alali et al. (2018) proposes using a Fuzzy Inference Model (FIS) to assess risk. It should consider four factors: vulnerability, threat, likelihood, and impact. The paper lacks detail on adversarial tactics, which this paper will cover. Insights from Haripriya and Kulothungan (2019) propose a novel IDS, Secure-MQTT, for MQTT-based IoT. It uses fuzzy logic to find any malicious devices. We use real-world data from MITRE ATT&CK to test attacks and their countermeasures.

2.4 Risk assessment-based methodology

EBIOS risk manager (de la Sécurité des Systémes d'Information, 2019) only offers a limited description of assets and risk scenarios. In this paper, we propose improvements by combining the complexity resilience index and detailed CVE metrics for better risk determination. We also explained detailed mitigation controls. Paper (Wu et al., 2023) introduces a risk assessment model using a Gini coefficient-based, evidence-reasoning approach rooted in Dempster-Shafer theory. The model addresses essential risk factors for cloud service providers integrating with diverse entities. However, its framework requires alignment with the advancements in Huang et al. (2024) to ensure its relevance against evolving cyber threats and dynamic business environments. Additionally, Casola et al. (2024) and Mills et al. (2023) propose a secure software development method tailored to modern DevOps pipelines, demonstrating its feasibility through a microservice application case study. Building on these works, our paper emphasizes the need for a comprehensive, multi-attribute risk assessment to enhance secure development methodologies.

2.5 Mitigation strategies

Recent studies show that most of the existing mitigation attempts for containers have drawbacks. For example, the Linux-based mitigation strategies used in containers, such as groups, namespaces, and capabilities, are prone to attacks due to resource exploitation, denials of services, and privilege escalation (Gao et al., 2019). Investigation from Wong et al. (2023); Dissanayaka et al. (2020) offers existing mitigation strategies and their limitations in a qualitative approach. A study from Devi Priya et al. (2023) proposing mitigation strategies from the DREAD threat modeling framework. A study from Koksal et al. (2024) attempts to conduct mitigation limited to DDoS attacks in container-based cloud environments using Kubernetes. We improve the mitigation efforts from the beginning of the paper by implementing a domain mapping matrix to ensure compliance with industry best practices and by demonstrating more attack vectors based on adversarial techniques and measurable quantitative efforts to reduce the risk scale from each impact.

3 Assets domain and adversarial techniques

A unified framework categorizes assets by their traits, limits, complexity, and sensitivity. This aids in systematic evaluations, as suggested by Kure et al. (2022); Assump cão et al. (2022). Figure 1 shows our main contribution to the paper's multi-attribute risk assessment. We adopted a methodology that employs modified fuzzy AHP, domain mapping matrix, and fuzzy logic. It improves the SSDE (Security SLA-based Security-by-Design Development) methodology by Casola et al. (2024).

3.1 Assets domain

The determination of system characterization begins with its asset identification. We enhance the provision of comprehensive asset descriptions based on previous publications (Hersyah et al., 2023; Blaise and Rebecchi, 2022). This paper identifies 5 (five) minimum assets and components that compose the basic foundation of MFA and AWS-labeled container orchestration utilization.

1. **MFA and authentication assets**: AWS Identity and Access Management (IAM) is the foundation. It configures MFA users, groups, roles, and permissions to ensure controlled access. AWS Cognito complements this. It enables user pool integration and offers risk scoring and verification. IAM users link to hardware or virtual MFA devices.
2. **Kubernetes assets**: include Amazon Elastic Kubernetes Service (EKS) clusters as the master components, AWS EC2 or Fargate as worker nodes, and AWS Elastic Block Store (EBS) for scalable, high-performance block storage.
3. **Docker container assets**: Include Amazon ECR for storing, managing, and deploying Docker images. Also, AWS CodeBuild is used to compile code, run tests, and produce deployable software.
4. **Cloud infrastructure assets**: include AWS Virtual Private Cloud (VPC) for isolating resources, subnets, and security groups for network segmentation and access control, and AWS Key Management Service (KMS) for encrypting data at rest and in transit. CloudTrail and CloudWatch support monitoring. They capture logs from Kubernetes, Docker, and other services. Load balancers like ALB and NLB manage traffic. AWS Route 53 is a DNS service that routes traffic to apps and services.
5. **Compliance and governance assets**: include AWS Config for monitoring resource configurations and ensuring compliance with defined rules, and AWS Security Hub for providing a centralized view of security best practices and compliance status across AWS accounts to detect.

3.2 Adversarial techniques

MFA, container engines, and orchestrators have exploitable flaws. Their reliance on software and hardware layers adds modularity. But it creates new attack surfaces. Critical security issues often arise from internal threats (Mahavaishnavi et al., 2024), misconfigurations (Renaud et al., 2024), and interdependencies (Bracke et al., 2024), leading to vulnerabilities that adversaries leverage to target deployed applications. We categorize adversarial techniques in Section 5.2 to evaluate these vulnerabilities. We analyze their actions using MITRE ATT&CK (MITRE Corporation, 2024). It is a framework that maps adversarial tactics and techniques from initial access to impact. The following details describe the adversarial techniques and their corresponding groups.

1. **Data destruction - adversarial group: APT38**: stemming from unauthorized external access or internal threats. Vulnerabilities include inadequate MFA, misconfigured Kubernetes, and insecure Docker images. These interdependencies can lead to data loss and business disruption.
2. **Endpoint denial of service - adversarial group: sandworm team**: malicious attacks involve using exposed APIs, unpatched Kubernetes or Docker software, and misconfigurations in

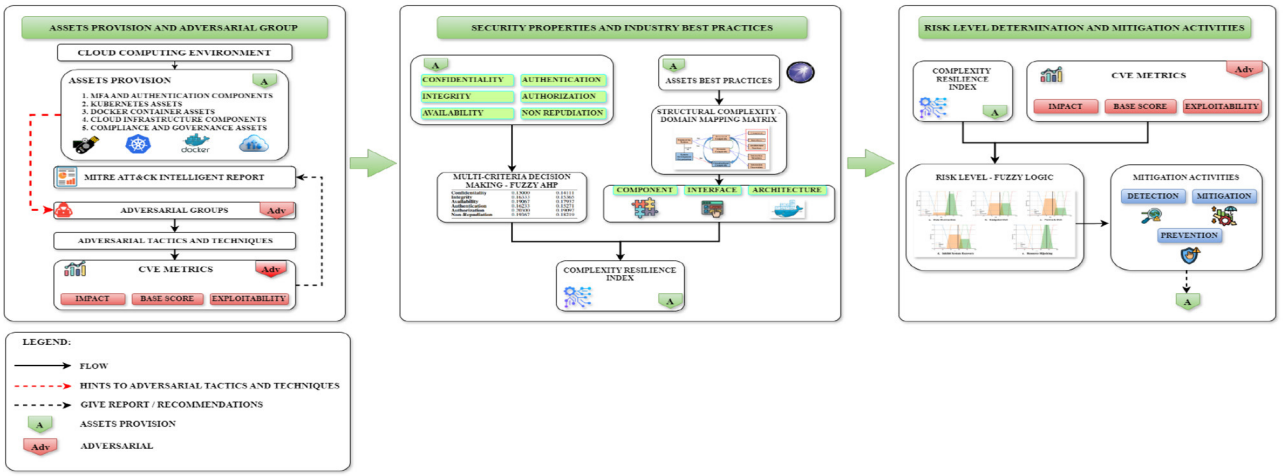

FIGURE 1
The proposed multi-attribute risk assessment.

service dependencies. resulting in operational downtime and potential loss of customer trust.

3. **Network denial of service - adversarial group: APT28**: exploiting weak network security, misconfigured policies, and interdependent systems in containerized environments. These attacks disrupt services and may result in revenue loss.
4. **Inhibit system recovery - adversarial group: wizard spider**: stemming from internal threats that conduct ransomware. Vulnerabilities and interdependencies arise from compromised container images and weak backup plans. They cause major financial losses, data loss, and high recovery costs.
5. **Resource hijacking - adversarial group: TeamTNT**: involving unauthorized use of cloud resources for malicious activities like cryptocurrency mining. Vulnerabilities include insecure Docker containers, weak Kubernetes authentication, and interdependent resource management systems. They lead to higher costs, lower performance, and compliance issues.

4 Methodology

This section explains the Modified Fuzzy AHP, Domain Mapping Matrix, and Fuzzy Logic. The Modified Fuzzy AHP is used to rank security priorities based on multiple criteria. The Domain Mapping Matrix identifies relationships between asset domains and aligns them with best practices. Fuzzy Logic is then applied to calculate the overall risk level.

4.1 Modified fuzzy AHP

Zadeh introduced fuzzy set theory in 1965 (Zadeh, 1965), laying the foundation for this technique, which is further explained in Emrouznejad and Ho (2017), where the integration of fuzzy logic into decision-making frameworks such as the Analytic Hierarchy Process (AHP) enables handling of uncertainty and vagueness in human judgments. In this study, we adopt and modify the Fuzzy AHP method developed by Chang (1996). We engaged four certified professionals in cloud security and container orchestration. Each expert was asked to conduct a pairwise comparison of the six CIA-AAN criteria using Saaty's 1–9 fundamental scale. To synthesize these individual judgments, we applied the Aggregated Mean Approach (Forman and Peniwati, 1998), and the arithmetic mean of each corresponding pairwise element across the expert matrices was calculated. This method, a standard form of Aggregation of Individual Judgments (AIJ), yields a central consensus matrix that reflects the collective view of the expert group (Tran et al., 2024). It ensures that no single expert dominates the evaluation and simplifies the fuzzification process.

Let:

- $A_k = [a_{ij}^{(k)}]$ be the pairwise comparison matrix provided by expert k, where $a_{ij}^{(k)}$ is the judgment of criterion i relative to criterion j from expert k.
- n is the number of criteria (e.g., 6 for CIA-AAN).
- K is the number of experts (e.g., 4 in this study).

Then, the consensus matrix $A = [\bar{a}_{ij}]$ is computed as follows:

$$\bar{a}_{ij} = \frac{1}{K} \sum_{k=1}^{K} a_{ij}^{(k)}, \quad \text{for } i,j = 1,2,\ldots,n \tag{1}$$

That is, each element of the consensus matrix is the arithmetic mean of all experts' judgments for the corresponding pairwise comparison:

$$\bar{a}_{ij} = \frac{a_{ij}^{(1)} + a_{ij}^{(2)} + \cdots + a_{ij}^{(K)}}{K} \tag{2}$$

After forming the consensus matrix, we applied Triangular Fuzzy Numbers (TFNs) to reflect the inherent uncertainty in expert judgments, as shown in Figure 2.

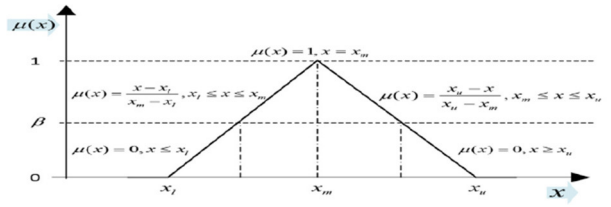

FIGURE 2
Triangular fuzzy number.

Step1: After using the fuzzy number operational laws, a fuzzy pairwise comparison matrix is given:

$$\tilde{X} = \begin{bmatrix} (1,1,1) & \tilde{a}_{12} & \cdots & \tilde{a}_{1n} \\ \tilde{a}_{21} & (1,1,1) & \cdots & \tilde{a}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{a}_{n1} & \tilde{a}_{n2} & \cdots & (1,1,1) \end{bmatrix},$$

$$V(M_1 \geq M_2) = \begin{cases} 1, & \text{if } m_2 \geq m_1, \\ 0, & \text{if } l_1 \geq u_2, \\ \frac{l_1 - u_1}{(m_2 - u_2) - (m_1 - u_1)}, & \text{otherwise.} \end{cases} \quad (3)$$

Step 2: The fuzzy geometric mean value $\tilde{r}_i$, for each criterion i is computed as

$$\tilde{r}_i = (\tilde{a}_{i1} \times \tilde{a}_{i2} \times \cdots \times \tilde{a}_{in})^{1/n} \quad (4)$$

Step 3: The fuzzy weight $\tilde{w}_i$ for each criterion i is calculated as

$$\tilde{w}_i = \tilde{r}_i \times (\tilde{r}_1 + \tilde{r}_2 + \cdots + \tilde{r}_n)^{-1}, \quad \text{where } \tilde{r}_k = (l_k, m_k, u_k)$$
$$\text{and } (\tilde{r}_k)^{-1} = \left(\frac{1}{u_k}, \frac{1}{m_k}, \frac{1}{l_k}\right). \quad (5)$$

Step 4: The technique is resumed by conducting de-fuzzification by formulating the center of Area (CoA)

$$\text{CoA}(\tilde{A}) = \frac{l + m + u}{3} \quad (6)$$

Step 5: The normalized weight vector to compute all components is = 1 (one).

$$NW_i = \frac{W_i}{\sum W_i} \quad (7)$$

4.2 Domain mapping matrix

We use a Domain mapping matrix (Maurer and Lindemann, 2008) to map elements between assets (MFA, Docker, and Kubernetes) and their best practices. It is a (l, m) rectangular binary adjacency matrix, where each entry indicates whether a specific best practice is applicable to a given asset.

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ A_{21} & A_{22} & \cdots & A_{2m} \\ \vdots & \vdots & A_{ij} & \vdots \\ A_{i1} & A_{i2} & \cdots & A_{lm} \end{bmatrix}, \quad A_{ij} = \begin{cases} 1, & \text{if requirement } i \text{and} \\ & \text{element } j \text{ have a relation,} \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

4.3 Fuzzy logic and Fuzzy Inference System description

Fuzzy Logic, implemented through a Fuzzy Inference System (FIS), provides a structured approach for reasoning under uncertainty and imprecision. This study employs the *skfuzzy* library (Warner, 2022) in Python to develop the FIS. The typical process involves four main stages (Geramian et al., 2019):

4.3.1 Fuzzification stage

In the fuzzification stage, precise input values are translated into fuzzy values. These fuzzy values are grouped into categories like low, medium, high, and very high. Each category is represented using a shape called a membership function (MF), which helps determine how strongly a value belongs to that category. Common shapes include triangular, trapezoidal, and Gaussian. In this study, we use trapezoidal membership functions, defined as:

$$\mu_{\text{trapezoidal}}(x; a, b, c, d) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right) \quad (9)$$

4.3.2 Fuzzy rule base stage

The Fuzzy Rule Base defines the relationship between input and output variables in this stage. Rules are expressed in an if-then format, where the antecedent (if part) describes the input conditions, and the consequent (then part) specifies the corresponding output action. Logical operators such as AND and OR combine the antecedent terms. These operators are mathematically represented as:

- **AND operation (Minimum method):** is used to model the intersection of fuzzy sets, where the membership degree of the combined condition is determined by taking the minimum value.

$$\mu_{\text{AND}} = \min(\mu_x, \mu_y) = \min(0.6, 0.8) = 0.6$$

- **OR operation (Maximum method):** is used to model the union of fuzzy sets, where the membership degree of the combined condition is determined by taking the maximum value.

$$\mu_{\text{OR}} = \max(\mu_x, \mu_y) = \max(0.6, 0.8) = 0.8$$

- **Combined use of AND and OR in rules**

  In Fuzzy Logic rules, AND and OR operators are often combined to handle complex relationships and shows flexibility. For example, consider the following rule:

  IF (n_1 is high OR n_2 is medium) AND (n_3 is high OR n_4 is medium)THEN *Output* is high.

  This rule uses both OR and AND operations:

  IF min (max($\mu_{n_1,\text{high}}, \mu_{n_2,\text{medium}}$), max($\mu_{n_3,high}, \mu_{n_4,medium}$))THEN *Output* is High.

4.3.3 Fuzzy inference and aggregation stage

In this stage, the defined rules are evaluated using the fuzzified input values, and the results are aggregated. Rule evaluation uses methods like the above mentioned methods, such as the AND and OR operations. After all rules are evaluated, their results are aggregated. Aggregation methods commonly used include:

- **Maximum method:**

$$\mu_{\text{aggregated}}(z) = \max(\mu_{\text{rule1}}(z), \mu_{\text{rule2}}(z), \dots) \quad (10)$$

4.3.4 Defuzzification stage

Finally, the aggregated output is defuzzified to obtain a crisp value. This is where fuzzy logic principles are translated into a precise numerical output. In this study, the discrete centroid method is employed:

$$z^* = \frac{\sum_{i=1}^{n} z_i \cdot \mu_{\text{aggregated}}(z_i)}{\sum_{i=1}^{n} \mu_{\text{aggregated}}(z_i)} \quad (11)$$

5 Proposed multi-attribute risk assessment

This section demonstrates the calculation stepwise of the proposed multi-attribute risk assessment described in Figure 1 as a proofing concept toward the contributions in Section 1.

5.1 Asset based assessment methods

5.1.1 Modified Fuzzy AHP

We structured the Fuzzy AHP process step by step in Section 4.1, beginning with the collection of objective expert judgments from four certified Kubernetes professionals. The detailed responses are provided in Supplementary Tables S4–S8. The aggregation of these expert inputs into consensus values is formalized in Equation 1, while the construction of the aggregated pairwise comparison matrix is outlined in Equation 2. The final comparison matrix evaluating the CIA-AAN security elements is presented in Table 1.

In the fuzzification process, crisp values from the traditional AHP scale (e.g., 1 to 9) were converted into triangular fuzzy numbers. For instance, a value of 1 was transformed into the fuzzy number (1,1,1), while a value of 4 was mapped to (3,4,5). Similarly, reciprocal values, such as 1/4, were converted into (1/5,1/4,1/3). We constructed the CIA-AAN fuzzy pairwise comparison matrix based on the judgments and applied fuzzification, using Equation 3, displayed in Table 2.

We computed geometric mean of each object element using Equation 4 displayed in Table 3:

We further compute by adding each row of Calculation Result of lower bound ($0.714 + 0.728 + 0.858 + 0.693 + 0.953 + 0.890$), Calculation Result of middle bound ($0.849 + 0.934 + 1.069 + 0.890 + 1.177 + 1.122$), and Calculation Result of upper bound ($1.049 + 1.164 + 1.371 + 1.200 + 1.399 + 1.348$), resulting the geometric mean value in 4.836, 6.041, and 7.531.

The next step is determining each criterion's fuzzy weight and Center of Area (CoA). We formulate the Fuzzy weight by multiplying the Calculation Result from Table 3 and the reciprocal values of the Geometric mean value (7.531, 6.041, 4.836), based on Equation 5. We compute defuzzification by formulating the Center of Area (CoA) using Equation 6 to give a crisp value. It is the average of its lower, middle, and upper parameters of the Fuzzy Weight, displayed in Table 4.

Finally, We get the normalized value from each criterion using the Equation 7 in Table 5. The normalized values of CIA-AAN serve as the security properties rank, which we will incorporate later with structural complexity to propose the complexity resilience index.

5.1.2 Domain mapping matrix

We assemble a domain mapping matrix as explained in Section 4.2 to improve the EBIOS risk manager. It maps assets to their best practices to describe structural complexity. It will assess asset value by comparing the assets with the FIDO2 best practices for MFA and the OWASP best practices for Kubernetes, Docker, and Cloud Computing (FIDO, 2022; OWASP, 2024c,b,a). We implement the Domain Mapping Matrix by adhering to Equation 8 in Table 6. The articulation assets consist of MFA and container orchestration in cloud environments. They total 16 in the column axis. We map them to 31 best practices from FIDO2 and OWASP in the row axis. These details include MFA, Kubernetes, Docker, and Cloud Computing best practices. We assign a value of **1** for a direct correlation between an asset and its best practices. In the absence of any identified direct relationship, we assign a value of **0**.

We apply Singular Value Decomposition (Σ), which reduces data dimensionality and generalizes the eigen decomposition for *mxn* matrix. It does this by extending the polar decomposition. It can be applied to multi-attribute risk assessment. Where Σ is an $m \times n$ *diagonal matrix*. It contains the singular values of A with a stretch nature, in 31 x 16. Only the first 16 rows would have non-zero values in the matrix's columns. The singular values, σ_i, come from the eigenvalues of A^TA (or AA^T, depending on

TABLE 1 Pairwise comparison matrix of CIA-AAN element properties.

	Confidentiality	Integrity	Availability	Authentication	Authorization	Non-repudiation
Confidentiality	1	1	1/4	1	3	1/2
Integrity	1	1	4	2	1/4	1/3
Availability	4	1/4	1	1/2	1	3
Authentication	1	1/2	2	1	1/2	1
Authorization	1/3	4	1	2	1	1
Non-Repudiation	2	3	1/3	1	1	1
Sum	9.33	9.75	8.58	7.5	6.75	6.83

TABLE 2 The CIA-AAN fuzzy pairwise comparison matrix.

	Confidentiality	Integrity	Availability	Authentication	Authorization	Non-repudiation
Confidentiality	(1, 1, 1)	(1, 1, 1)	(1/5, 1/4, 1/3)	(1, 1, 1)	(2, 3, 4)	(1/3, 1/2, 1)
Integrity	(1, 1, 1)	(1, 1, 1)	(3, 4, 5)	(1, 2, 3)	(1/5, 1/4, 1/3)	(1/4, 1/3, 1/2)
Availability	(3, 4, 5)	(1/5, 1/4, 1/3)	(1, 1, 1)	(1/3, 1/2, 1)	(1, 1, 1)	(2, 3, 4)
Authentication	(1, 1, 1)	(1/3, 1/2, 1)	(1, 2, 3)	(1, 1, 1)	(1/3, 1/2, 1)	(1, 1, 1)
Authorization	(1/4, 1/3, 1/2)	(3, 4, 5)	(1, 1, 1)	(1, 2, 3)	(1, 1, 1)	(1, 1, 1)
Non-Repudiation	(1, 2, 3)	(2, 3, 4)	(1/4, 1/3, 1/2)	(1, 1, 1)	(1, 1, 1)	(1, 1, 1)

TABLE 3 Fuzzy geometric mean calculation and results for each criterion.

Criterion	Geometric mean calculation	Lower bound	Middle bound	Upper bound
Confidentiality	$(1\times1\times\frac{1}{5}\times1\times2\times\frac{1}{3})^{\frac{1}{6}}, (1\times1\times\frac{1}{4}\times1\times3\times\frac{1}{2})^{\frac{1}{6}}, (1\times1\times\frac{1}{3}\times1\times4\times1)^{\frac{1}{6}}$	0.714	0.849	1.049
Integrity	$(1\times1\times3\times1\times\frac{1}{5}\times\frac{1}{4})^{\frac{1}{6}}, (1\times1\times4\times2\times\frac{1}{4}\times\frac{1}{3})^{\frac{1}{6}}, (1\times1\times5\times3\times\frac{1}{3}\times\frac{1}{2})^{\frac{1}{6}}$	0.728	0.934	1.164
Availability	$(3\times\frac{1}{5}\times1\times\frac{1}{3}\times1\times2)^{\frac{1}{6}}, (4\times\frac{1}{4}\times1\times\frac{1}{2}\times1\times3)^{\frac{1}{6}}, (5\times\frac{1}{3}\times1\times1\times1\times4)^{\frac{1}{6}}$	0.858	1.069	1.371
Authentication	$(1\times\frac{1}{3}\times1\times1\times\frac{1}{3}\times1)^{\frac{1}{6}}, (1\times\frac{1}{2}\times2\times1\times\frac{1}{2}\times1)^{\frac{1}{6}}, (1\times1\times3\times1\times1\times1)^{\frac{1}{6}}$	0.693	0.890	1.200
Authorization	$(\frac{1}{4}\times3\times1\times1\times1\times1)^{\frac{1}{6}}, (\frac{1}{3}\times4\times1\times2\times1\times1)^{\frac{1}{6}}, (\frac{1}{2}\times5\times1\times3\times1\times1)^{\frac{1}{6}}$	0.953	1.177	1.399
Non-repudiation	$(1\times2\times\frac{1}{4}\times1\times1\times1)^{\frac{1}{6}}, (2\times3\times\frac{1}{3}\times1\times1\times1)^{\frac{1}{6}}, (3\times4\times\frac{1}{2}\times1\times1\times1)^{\frac{1}{6}}$	0.890	1.122	1.348

the dimensions). The singular values are the square roots of these eigenvalues, which are defined as follows:

$$\sigma_i = \sqrt{\lambda_i} \tag{12}$$

where λ_i are the eigenvalues of A^TA (or AA^T). The Singular Value of A is given by:

$$\begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \\ \vdots \\ V_r^T \\ V_{r+1}^T \\ \vdots \\ V_n^T \end{bmatrix} = \begin{bmatrix} \text{Row } A \\ \text{Null } A \end{bmatrix} \tag{13}$$

where Σ_1, Σ_2, …, Σ_n are the singular values. These are non-negative and are typically arranged in descending order. The formulation of singular value is as follows:

$$\Sigma = \begin{bmatrix} 9.2498 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5.2087 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.7101 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.5898 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.4903 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.0203 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.4057 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.3579 \end{bmatrix}$$

$$\begin{bmatrix} 1.1421 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9265 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.6504 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.3867 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0000 \end{bmatrix}$$

We referenced paper Sheard and Mostashari (2009) and Sinha and Suh (2018) to calculate structural complexity metrics, which

TABLE 4 Fuzzy weight and center of area (CoA).

	Values	Fuzzy weight	CoA
Confidentiality	$(0.714, 0.849, 1.049) \otimes (1/7.531, 1/6.041, 1/4.836)$	$(0.094, 0.140, 0.216)$	$(0.094 + 0.140 + 0.216)/3 = \mathbf{0.15000}$
Integrity	$(0.728, 0.934, 1.164) \otimes (1/7.531, 1/6.041, 1/4.836)$	$(0.096, 0.154, 0.240)$	$(0.096 + 0.154 + 0.240)/3 = \mathbf{0.16333}$
Availability	$(0.858, 1.069, 1.371) \otimes (1/7.531, 1/6.041, 1/4.836)$	$(0.113, 0.176, 0.283)$	$(0.113 + 0.176 + 0.283)/3 = \mathbf{0.19067}$
Authentication	$(0.693, 0.890, 1.200) \otimes (1/7.531, 1/6.041, 1/4.836)$	$(0.092, 0.147, 0.248)$	$(0.092 + 0.147 + 0.248)/3 = \mathbf{0.16233}$
Authorization	$(0.953, 1.177, 1.399) \otimes (1/7.531, 1/6.041, 1/4.836)$	$(0.126, 0.194, 0.289)$	$(0.126 + 0.194 + 0.289)/3 = \mathbf{0.20300}$
Non-repudiation	$(0.890, 1.122, 1.348) \otimes (1/7.531, 1/6.041, 1/4.836)$	$(0.118, 0.185, 0.278)$	$(0.118 + 0.185 + 0.278)/3 = \mathbf{0.19367}$

TABLE 5 The CIA-AAN normalized values.

	Center of area (CoA)	Normalized values	Priority (%)
Confidentiality	0.15000	0.15000/ 1.063 = **0.14111**	14.111%
Integrity	0.16333	0.16333/ 1.063 = **0.15365**	15.365%
Availability	0.19067	0.19067/ 1.063 = **0.17937**	17.937%
Authentication	0.16233	0.16233/ 1.063 = **0.15271**	15.271%
Authorization	0.20300	0.20300/ 1.063 = **0.19097**	19.097%
Non-repudiation	0.19367	0.19367/ 1.063 = **0.18219**	18.219%
Accumulated value	**1.063**	**1.0**	**100.00%**

consist of components, interfaces, and architecture. Based on this, we proposed a complexity resilience index using the proposed Equations 17–20.

5.1.3 Structural complexity 1: components

This aspect is related to component engineering. The singular value decomposition formula gives us the singular values of the 16 assets: 9.2498, 5.2087, 3.7101, 2.5898, 2.4903, 2.0203, 1.4057, 1.3579, 1.1421, 0.9265, 0.6504, 0.3867, 0.0000, 0.0000, 0.0000, 0.0000. And for a variable c defined as the sum of the first k singular values from the domain mapping matrix:

$$C_1 = \sum_{i=1}^{k} \sigma_i, \quad \text{where} \begin{cases} C_1 \text{ denotes the component score,} \\ \sigma_i \text{ represents the singular value associated} \\ \text{with the asset } i. \end{cases} \tag{14}$$

This represents the accumulation of singular values that contribute to the component metric of structural complexity. The sum of the singular values is given by:

$$\begin{aligned} C_1 = {} & 9.2498 + 5.2087 + 3.7101 + 2.5898 + 2.4903 + 2.0203 \\ & + 1.4057 + 1.3579 + 1.1421 + 0.9265 + 0.6504 + 0.3867 \\ & + 0.0000 + 0.0000 + 0.0000 + 0.0000 = \mathbf{31.1382} \end{aligned}$$

5.1.4 Structural complexity 2: interfaces

The second aspect is related to interface design and management, which is the cumulative term that explains interaction complexity β_{ij} between components, which we expressed as the following formula:

$$C_2 = \sum_{i=1}^{k} \sum_{j=1}^{k} \beta_{ij} A_{ij}, \quad A_{ij} = \begin{cases} 1, & \forall[(i,j)|(i \neq j)] \in A, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

The cumulative sum from the domain mapping matrix, which we can find in the last row of Table 6, is the total of the granular assets score, which is given by:

$$\begin{aligned} C_2 = {} & 7 + 6 + 6 + 23 + 18 + 7 + 27 + 11 + 9 + 8 + 6 + 9 + 1 \\ & + 1 + 3 + 8 = \mathbf{150} \end{aligned}$$

5.1.5 Structural complexity 3: architecture

The last aspect is related to the system integration effort to address the architecture topology metric, which we expressed in the following formula:

$$C_3 = \frac{\sum_{i=1}^{k} \sigma_i}{\min(l, m)}, \quad \text{where} \begin{cases} \sum_{i=1}^{k} \sigma_i \text{ is the Component Aspect,} \\ l \text{ is the number of best practices,} \\ m \text{ is the number of asset domains.} \end{cases} \tag{16}$$

The architecture metric can be obtained as follows:

$$C_3 = \frac{31.1382}{16} = \mathbf{1.9461}$$

5.1.6 Complexity resilience index determination

We propose the complexity resilience index by formulating linear computation between the structural complexity metrics (component, interface, and architecture) and the normalized values of CIA-AAN from the modified fuzzy AHP in Table 5.

1. **Component with availability and non-repudiation**: these tools provide operations to manage and scale applications across diverse environments. Emphasizing availability ensures the app works well and is always accessible. It also minimizes downtime (Alahmad et al., 2019). Also, non-repudiation means logging all system actions. This includes container deployments and Kubernetes changes. It provides undeniable accountability (Truyen et al., 2020).

TABLE 6 Domain mapping matrix.

		ASSETS DOMAIN	MFA AUTHENTICATION			KUBERNETES			DOCKER		CLOUD INFRASTRCUTURE						GOVERNANCE AND MONITORING	
		ASSETS NAME	AWS IAM	AWS COGNITO	MFA DEVICE	AWS EKS	K8S WORKER NODES (EC2/Fargate)	AWS ELASTIC BLOCK	AWS ELASTIC CONTAINER REGISTRY	AWS CODEBUILDER	AWS VPC	SUBNETS AND SECURITY GROUP	AWS KMS	CLOUDWATCH AND CLOUDTRAIL	LOAD BALANCER	AWS ROUTE 53	AWS CONFIG	AWS SECURITY
DOMAIN	BEST PRACTICE REQUIREMENTS	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
FIDO2 [FIDO2 Adoption]	Avoid Using Domain Hints to Bypass Home-Realm Discovery and account recovery using back up authenticators, recovery methods, cloud-based recovery	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1
	Ensure Compatibility with WebAuthn and CTAP2 Standards	2	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1
	Implement Attestation Enforcement and Key Restriction Policies by enforcing attestation ensures that only genuine FIDO2 security keys or passkey providers are used.	3	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1
	Enabling Phishing-Resistant Passwordless Authentication	4	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1
KUBERNETES HOSTS AND COMPONENTS [OWASP Kubernetes]	Update Kubernetes and secure its dashboard	5	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
	Control and Restrict network access to kubelets and etcd and sensitive ports	6	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0
	Controlling access to the Kubernetes API	7	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
KUBERNETES BUILD PHASE [OWASP Kubernetes]	Implementing role-based access control in Kubernetes	8	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
	Only use authorized images	9	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
	Use a CI Pipeline to control and identify Vulnerabilities.	10	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
	Minimize features in all CIs	11	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0
KUBERNETES DEPLOY PHASE [OWASP Kubernetes]	Code that uses namespaces to isolate Kubernetes resources	12	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0
	Use the imagePolicyWebhook to govern image provenance.	13	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0
	Implement continuous security by service mesh and vulnerability scanning	14	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0
	Continuously assess the privileges used by containers	15	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0
	Implement centralized policy management and limit resource usage	16	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
	Use Kubernetes network policies to control traffic between pods and clusters	17	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0
KUBERNETES RUNTIME PHASE [OWASP Kubernetes]	Use Pod security policies to prevent risky containers/pods	18	0	0	0	1	1	0	1	0	1	1	0	1	0	0	0	0
	Container runtime security and sandboxing	19	0	0	0	1	1	0	1	1	1	1	0	1	0	0	0	0
	Preventing containers from loading unwanted kernel modules	20	0	0	0	1	1	0	1	0	1	1	0	1	0	0	0	0
	Compare and analyze different runtime activities in pods of same deployments	21	0	0	0	1	1	0	1	1	1	1	0	1	0	0	0	0
	Monitor network traffic to limit unnecessary communication.	22	0	0	0	1	1	0	1	0	1	1	0	1	0	0	0	0
	If breached, scale suspicious pods to zero.	23	0	0	0	1	1	0	1	0	1	1	0	1	0	0	0	0
DOCKER [OWASP Docker]	Keep Host and Docker up to date and use Linux security module	24	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0
	Do not expose the docker daemon socket even at inter-container	25	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	Limit capabilities and resources and run in rootless mode	26	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	Prevent in-container privilege escalation	27	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
CLOUD COMPUTING [OWASP Cloud Computing]	Secure Object Storage	28	1	0	0	1	0	1	1	0	1	1	1	0	0	0	0	1
	Implement Zero Trust architecture and security tools	29	1	1	1	1	1	0	1	1	1	1	0	1	0	1	1	1
	Implement shared responsibility model	30	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1
	Encrypt Data in Transit and Rest	31	0	0	0	1	1	1	1	1	1	0	1	0	1	0	0	1
	TOTAL GRANULAR ASSET SCORE		7	6	6	23	18	7	27	11	9	8	6	9	1	1	3	8

2. **Interface with integrity, authentication, and authorization**: data in systems built with Docker and Kubernetes must be trustworthy and unchanged. So, we must maintain its integrity as it moves through interfaces. This setup uses FIDO2 to strengthen authentication. It ensures that only verified users and services can access the system. Authorization defines what authenticated users can do. It controls resource access based on policies (Kudo et al., 2021; Bánáti et al., 2018).
3. **Architecture with confidentiality**: confidentiality protects private, sensitive information from unauthorized access (Seifermann et al., 2019). This approach uses strong encryption and secure access controls. They protect sensitive data at rest and in transit.

To calculate the complexity resilience index score, we propose the following linear computation based on the security attributes assigned to each complexity aspect:

$$w_c = C_1 \times \left(\frac{1}{2} \times \sum_{i \in \{a,nr\}} w_i \right), \tag{17}$$

$$w_i = C_2 \times \left(\frac{1}{3} \times \sum_{j \in \{int,auh,autz\}} w_j \right), \tag{18}$$

$$w_a = C_3 \times \left(\sum_{k \in \{conf\}} w_k \right), \tag{19}$$

$$\text{complexity resilience index} = w_c + w_i + w_a. \tag{20}$$

We adopt the Cyclomatic Complexity Metric as defined by McCabe (McCabe, 1976), which is widely used to evaluate ranges of software complexity:

- 1 – 10: Simple procedure
- 11 – 20: Medium Procedure

- 21 – 50: Complex Procedure
- > 50: Untestable code

We computed all Equations from 17 to 20 to obtain proposed complexity resilience index scores.

$$w_c = 31.1382 \times \left(\frac{0.17937 + 0.18219}{2}\right) = 8.4218$$

$$w_i = 150 \times \left(\frac{0.15365 + 0.15271 + 0.19097}{3}\right) = 24.8665$$

$$w_a = 1.9461 \times (0.14111) = 0.2746$$

$$\text{complexity resilience index} = 8.4218 + 24.8665 + 0.2746 = 33.5629 \equiv 34 \textbf{ (Complex Procedure)}.$$

5.2 Adversarial technique-based assessment methods

This subsection examines adversarial techniques discussed in Section 3.2. These techniques are organized within adversarial

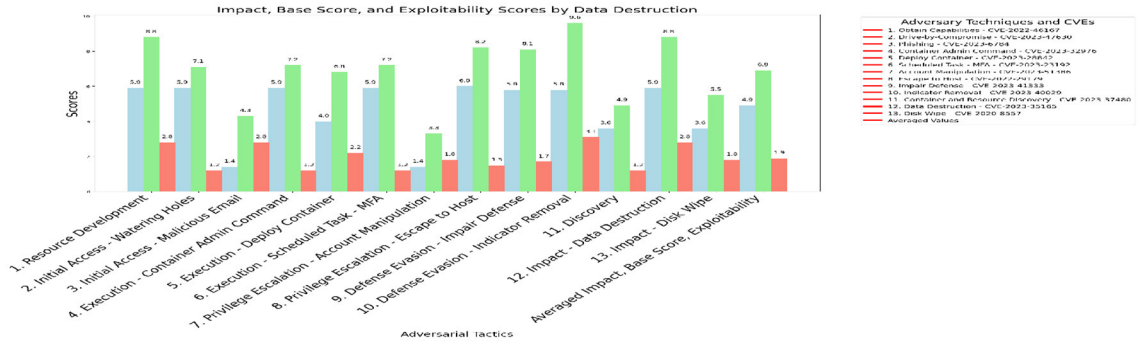

FIGURE 3
Data destruction impact, base score and exploitability.

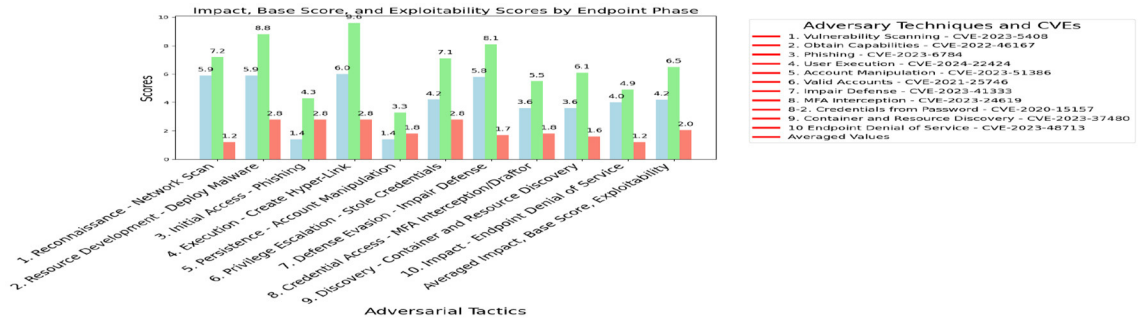

FIGURE 4
Endpoint DOS impact, base score, and exploitability.

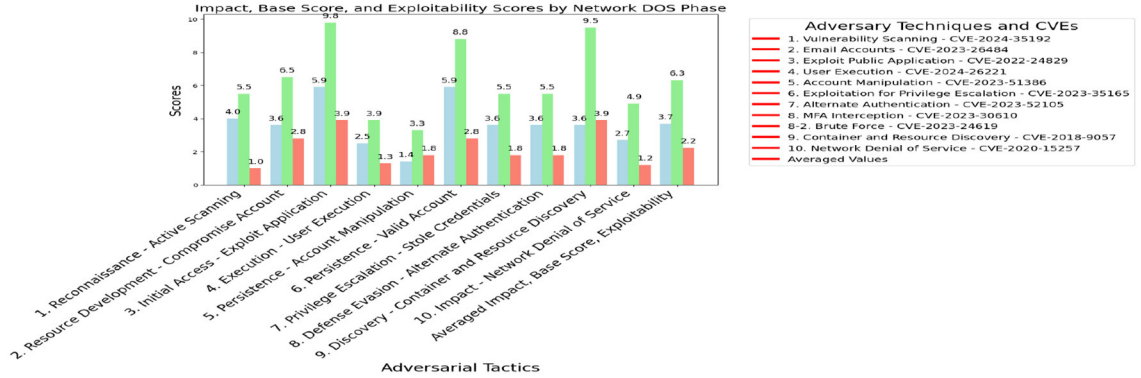

FIGURE 5
Network DOS impact, base score, and exploitability.

tactics that contain methods from external and internal threats, misconfigurations, and interdependencies targeting vulnerabilities in these systems. The evaluation focuses on the CVE's impact, base score, and exploitability associated with five critical techniques identified in the MITRE ATT&CK framework (MITRE Corporation, 2024): Data Destruction (T1485), Endpoint Denial of Service (DoS) (T1499), Inhibit System Recovery (T1490), Network Denial of Service (DoS) (T1498), and Resource Hijacking (T1496). These techniques span multiple stages of adversarial tactics, from initial access to impact, and are particularly critical due to their potential to disrupt FIDO2 for MFA and container orchestration systems in cloud environments. The analysis uses a stacked bar chart to show the CVE metrics. It highlights the impact (blue), base score (green), and exploitability (red).

5.2.1 Data destruction—adversarial group APT38

The APT38 group (MITRE ATT&CK, 2024b) uses CVE-2023-23192 to bypass authentication, which internal threats can also abuse. Misconfigurations, such as those associated with CVE-2023-28842, arise during phases like Execution–Deploy Container, where improperly secured configurations enable adversaries to deploy malicious containers. Interdependency issues, exemplified by CVE-2022-29179, often occur in the Privilege Escalation–Escape to Host phase, where weak interconnections between containerized environments and host systems are illustrated in

TABLE 7 Degree of membership functions for input, output variables.

Variables	Type	Range and limiter
Complexity resilience index	Trapezoidal	Low (1 1 8 15), Medium (10 15 18 25), High (20 25 45 55), Very High (50 55 100 100)
Impact	Trapezoidal	Low (0.1 0.1 3.0 3.9), Medium (4.0 4.0 6.0 6.9), High (7.0 7.0 8.5 8.9), Critical (9.0 9.0 10.0 10.0)
Base score	Trapezoidal	Low (0.1 0.1 3.0 3.9), Medium (4.0 4.0 6.0 6.9), High (7.0 7.0 8.5 8.9), Critical (9.0 9.0 10.0 10.0)
Exploitability	Trapezoidal	Low (0.1 0.1 3.0 3.9), Medium (4.0 4.0 6.0 6.9), High (7.0 7.0 8.5 8.9), Critical (9.0 9.0 10.0 10.0)
Risk level	Trapezoidal	Low (0 0 19.5 39), Medium (30 40 59 69), High (60 70 79.5 89), Critical (80 90 100 100)

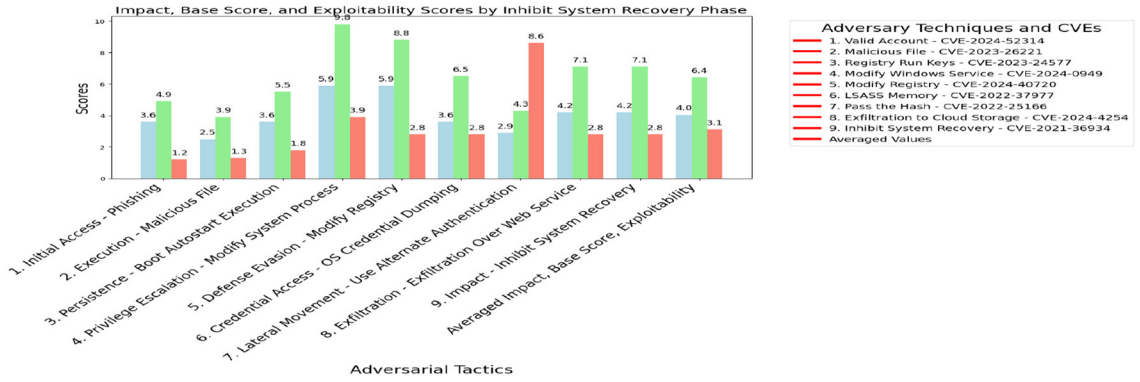

FIGURE 6
Inhibit system recovery impact, base score, and exploitability.

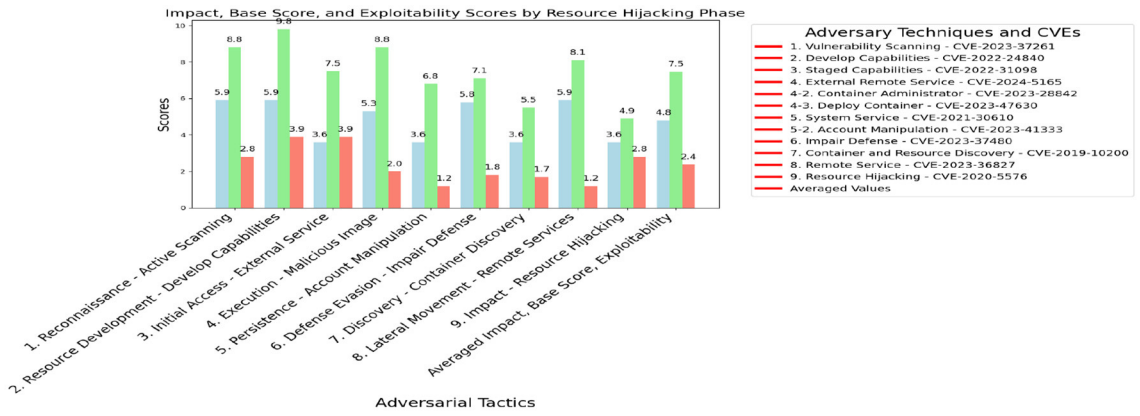

FIGURE 7
Resource hijacking impact, base score, and exploitability.

Figure 3. These vulnerabilities result in an average impact score of **4.9**, a base score of **6.9**, and an exploitability score of **1.9**.

5.2.2 Endpoint of denial services—adversarial group sandworm team

The Sandworm's cyber threat level, as outlined in MITRE ATT&CK (2024a). Sandworm employs CVE-2023-24619 for MFA interception and leveraging container orchestration vulnerabilities. Misconfigurations associated with CVE-2023-37480 are exploited during the Container Discovery phase, where inadequate configurations allow adversaries to exploit containerized environments. Interdependency issues, highlighted by CVE-2021-25746, occur during the Privilege Escalation–Stole Credentials phase, where weak interactions between containerized systems and authentication mechanisms enable unauthorized credential access. Figure 4 shows the impact of **4.2**, base score of **6.5**, and exploitability score of **2.0**.

TABLE 8 Fuzzy logic rules.

Rules	Configuration	Risk level
Rule 1	IF (Complexity Resilience Index is Low AND Impact is Low) OR (Base Score is Low AND Exploitability is Low)	Low
Rule 2	IF (Complexity Resilience Index is Medium OR Impact is Medium) OR (Base Score is Medium OR Exploitability is Medium)	Medium
Rule 3	IF Complexity Resilience Index is High AND Impact is Medium AND Base Score is High AND Exploitability is Low	Medium
Rule 4	IF Complexity Resilience Index is High AND Impact is High AND Base Score is High AND Exploitability is High	High
Rule 5	IF (Complexity Resilience Index is High OR Impact is High) AND (Base Score is High OR Exploitability is High)	High
Rule 6	IF (Complexity Resilience Index is Very High AND Impact is Critical) OR (Base Score is Critical AND Exploitability is Critical)	Catastrophic

5.2.3 Network denial of service—adversarial group APT28

Figure 5 examines the APT28 as outline in MITRE ATT&CK (2024a). This group used CVE-2023-52105 to bypass authentication. They also used CVE-2023-30610 to intercept MFA. These attacks targeted the authentication. Misconfigurations associated with CVE-2022-24829 were exploited during the Initial Access–Exploit Public Application phase, allowing attackers to compromise application environments. Interdependency issues linked to CVE-2018-9057 were identified in the Discovery–Container and Resource Discovery phase, where adversaries leveraged weak dependencies within containerized systems. Their average impact, base score, and exploitability: **3.7**, **6.3**, and **2.2**.

5.2.4 Inhibit system recovery—adversarial group wizard spider

The Spider group, as noted in MITRE ATT&CK (2024d), uses methods like LSASS memory dumping CVE-2022-37977

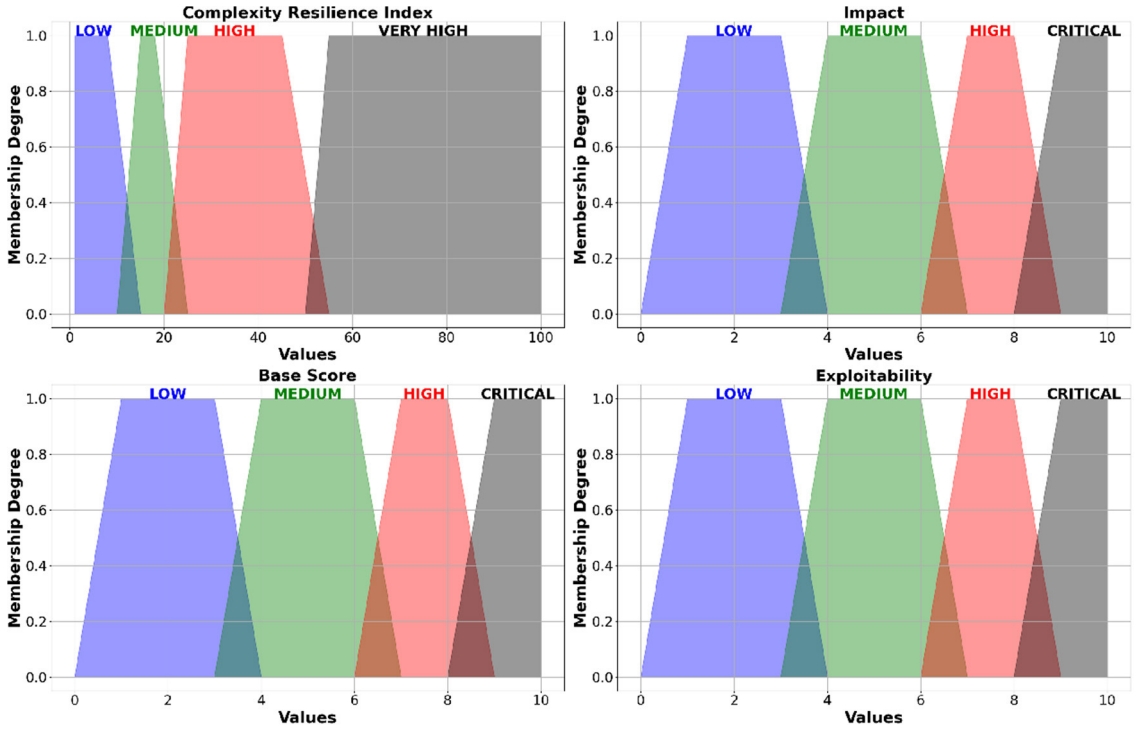

FIGURE 8
Input variables categorized for fuzzy logic assessment.

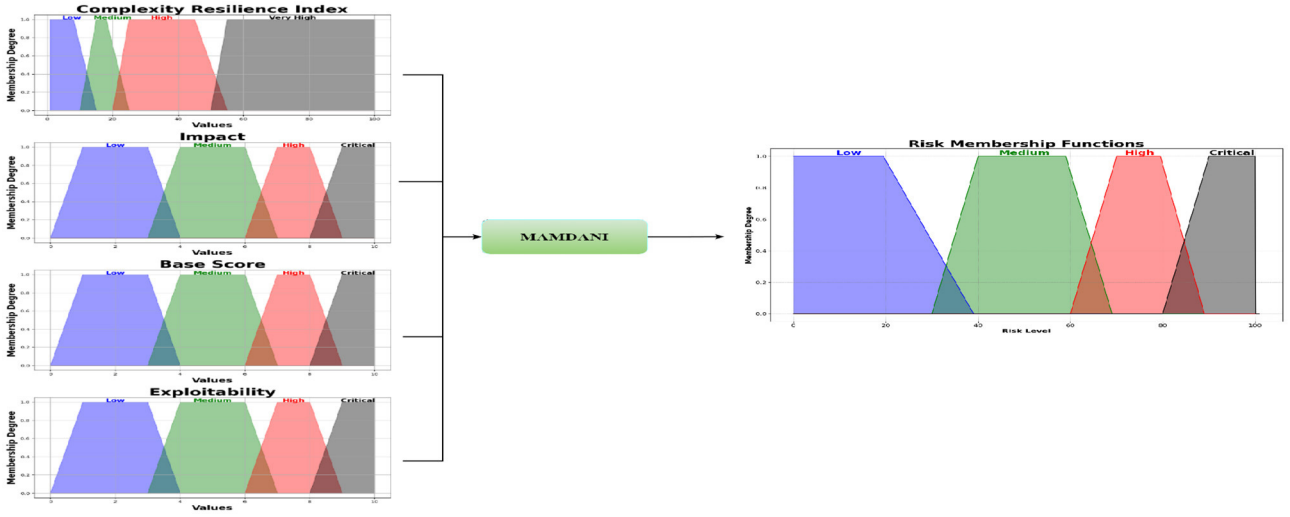

FIGURE 9
Fuzzy inference system plotting.

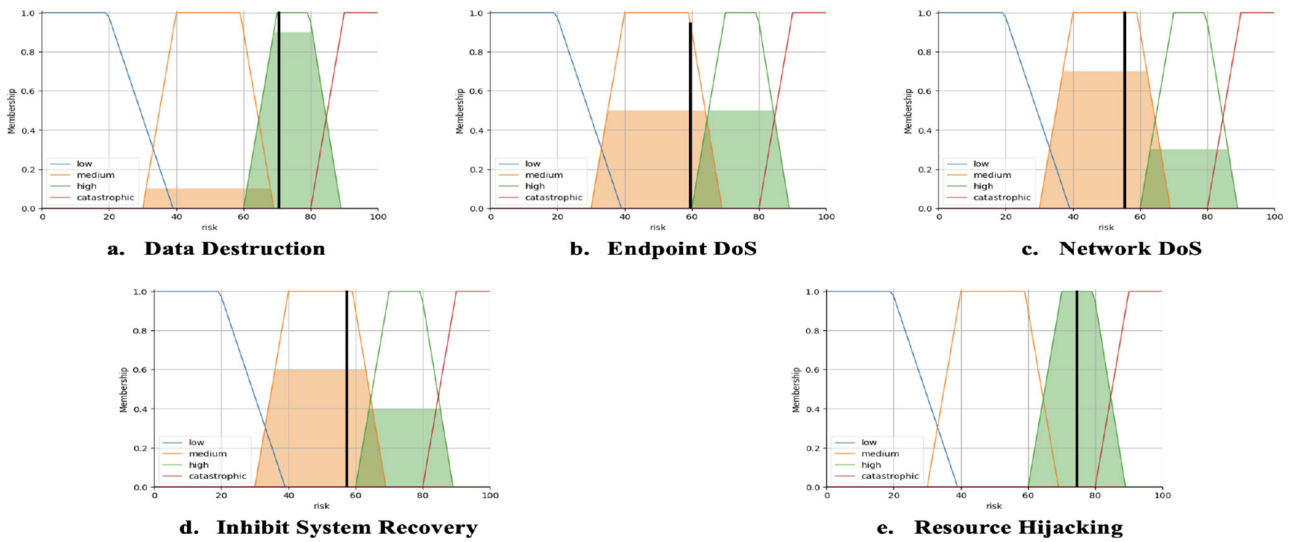

FIGURE 10
Risk assessment result. **(a)** Data destruction, **(b)** Endpoint DoS, **(c)** Network Dos, **(d)** Inhibit system recovery, **(e)** Resource hijacking.

TABLE 9 Risk assessment determination.

Adversarial techniques	Complexity resilience index	Impact	Base score	Exploitability	Risk score	Risk level
Data destruction	34	4.9	6.9	1.9	70.47	**High level**
Endpoint DOS	34	4.2	6.5	2.0	59.56	**Medium level**
Network DOS	34	3.7	6.3	2.2	55.44	**Medium level**
Inhibit system recovery	34	4.0	6.4	3.1	57.45	**Medium level**
Resource hijacking	34	4.8	7.5	2.4	74.60	**High level**

and Pass the Hash CVE-2022-25166 to break authentication. Misconfigurations like CVE-2024-40720 were exploited during the Defense Evasion–Modify Registry phase, where attackers altered critical registry settings to evade detection. Interdependency issues, linked to CVE-2021-36934, are observed during the Impact-Inhibit system Recovery phase, where weak system recovery protocols

TABLE 10 Proposed mitigation activities: data destruction.

Current risk score and level	Acceptable risk score and level	Critical state conducted from adversarial group	Detection	Mitigation	Prevention	Initial risk	Risk reduction	Residual risk
70.47 High	**<= 39.0 Low**	Resource development obtain capabilities	Internet scan, malware repository	Pre-compromise	Data governance	70.47	3%	68.36
		Initial access watering holes	Application log, network connection and traffic	Application isolation, update software	Regular data handling	68.36	5%	64.94
		Initial access malicious email	Network traffic content	Vulnerability scanning, network segmentation	Data governance	64.94	5%	61.69
		Execution container admin command	Command execution and process creation	Privileged account management	Screening	61.69	5%	58.61
		Execution deploy container	Monitor container creation and start, pod creation and modification	Audit, limit access to resource over network	Regular data handling	58.61	5%	55.68
		Execution scheduled task - mfa	Monitor container and file creation	Restrict file and directory permission, user account management	File integrity monitor	55.68	5%	52.89
		Privilege escalation account manipulation	User account modification, active directory object	Deploy mfa, network segmentation, user account management	Deploy user and behavior analytics	52.89	5%	50.25
		Privilege escalation escape to host	Monitor particular container running as root, kernel module load	Privileged account management, application isolation	Regular audit	50.25	5%	47.74
		Defense evasion impair defense	Monitor executed command and script, change in firewall status, monitor changes to cloud service	Audit, implement policies in software configuration, restrict registry permission	Monitor compliance with data retention	47.74	5%	45.35
		Defense evasion indicator removal	Monitor for api calls that may delete artifacts, user account authentication, monitor logs	Encrypt sensitive information, remote data storage	Regular data storage inspection	45.35	5%	43.08
		Discovery	Container and pod enumeration	Limit access to resource over network, network segmentation	Secure data backups	43.08	5%	40.93
		Impact data destruction	Monitor for unexpected modification and deletion in cloud including images, instance, snapshot	Backup regularly, mfa, user account management	Regular inspection of data storage	40.93	5%	38.88
		Impact disk wipe	Monitor for new process creation, command execution, drive access, and modification	Backup regularly	Update incident response and recovery plan	38.88	5%	36.94

10.3389/fcomp.2025.1557918

10.3389/fcomp.2025.1557918

TABLE 11 Proposed mitigation details: endpoint DoS.

Current risk score and level	Acceptable risk score and level	Critical state conducted from adversarial group	Detection	Mitigation	Prevention	Initial risk	Risk reduction	Residual risk
59.56 Medium	**<=39.0 Low**	Reconnaissance Network Scan	Monitor and analyze patterns and packets that don't follow protocol standards	Pre-Compromise	Apply rate limiting and Geo-blocking	59.56	3%	57.77
		Resource Development Obtain Capabilities - Deploy Malware	Analyze malware features and monitor for contextual data about malicious payload	Pre-Compromise	Use AWS Cloudwatch and Cloudtrail	57.77	3%	56.04
		Initial Access Phishing	Monitor for third-party application logging, newly constructed files, analyze SSL/TLS traffic	Implement anti-virus, Audit, Software configuration, and user training	Conduct training on identifying and responding to DoS	56.04	5%	53.24
		Execution User Execution - Hyperlink	Monitor for files created, network connections, and inspect the content of network traffic	User training, restrict web-based content	Policy enforcement on access control	53.24	5%	50.58
		Persistence Account Manipulation	Monitor events of accounts and its permissions, group and file modification, registration of new devices	User account management, removing potentially abused authentication and authorization	Policy enforcement on access control	50.58	5%	48.05
		Privilege Escalation Stole Credentials	Monitor for an attempt by the user that may abuse credentials, monitor new login behavior and its session metadata	Use conditional access to block logins from non-compliant devices, password policies, user account management	Apply rate limiting and Geo-blocking	48.05	5%	45.64
		Defense Evasion Impair Defense	Monitor logs for API calls to disable logging, monitor executed commands, monitor process creation/modification/termination	Use application control where appropriate, execution prevention, restrict registry permissions	Use AWS Cloudwatch and Cloudtrail to monitor logs	45.64	5%	43.36
		Credential Access MFA Interception	Monitor for proxied smart card, API calls, and changes to registries	Conduct user training and policy to remove peripherals when not in use	Regular audit, user training	43.36	5%	41.19
		Discovery Container and Resource Discovery	Monitor logs for actions taken to gather information about container infrastructure and pods including API calls	Limit communication with container service to secure channels, deny direct remote access through proxies, gateways, and firewalls, enforce the least privilege access rights	Policy enforcement on resource allocation	41.19	5%	39.13
		Impact Endpoint Denial of Service	Monitor for third-party logging, analyze traffic patterns, detection on host status	Filter network traffic by using services provided by Content Delivery Networks (CDN)	Develop incident response, including steps to isolate the affected endpoints	39.13	5%	37.18

TABLE 12 Proposed mitigation activities: network DoS.

Current risk score and level	Acceptable risk score and level	Critical state conducted from adversarial group	Detection	Mitigation	Prevention	Initial risk	Risk reduction	Residual risk
55.44 Medium	**< = 39.0 Low**	Reconnaissance Active Scanning - Vulnerability	Monitor for suspicious network traffic that indicates probing on user information	Pre-Compromise	Develop network security policies	55.44	3%	53.78
		Resource Development Compromise Account - Email	No Available Detection	Pre-Compromise	Use AWS Shield and WAF	53.78	1%	53.24
		Initial Access Exploit Application	Detect software exploitation in its application's logs, use deep packet inspection to look for common exploit traffic	Conduct vulnerability scanning, application isolation, use web application firewalls, and network segmentation	Implement real-time monitoring	53.24	5%	50.58
		Execution User Execution	Monitor logs from applications to detect user-initiated actions, identify processes spawned by user actions that could lead to malicious execution, monitor network traffic patterns, container and image creation, and newly operated	Behavior prevention on the endpoint, implement network intrusion prevention, restrict web-based content, and conduct user training	Develop policies for acceptable use, traffic patterns, and performance	50.58	5%	48.05
		Persistence Account Manipulation	Monitor events for changes to accounts and permissions, monitor for the registration of new device objects, executed commands, files, and group modification, and process creation	Implement privileged account management, restrict access to sensitive files that deal with authentication and authorization, configure access control, and protect domain controllers	Implement anomaly detection	48.05	5%	45.65
		Persistence Valid Account	Monitor for attempts by a user that abuses the credentials of existing accounts, monitor new login behavior, and look for suspicious behavior that shares accounts	Train users to accept valid and report suspicious notifications, audit domain local accounts, implement password policies and MFA, ensure applications don't store sensitive data, conditional access points to block logins from non-compliant devices and disable legacy authentication that does not support MFA	Apply rate limiting and Geo-Blocking, User Training, Audit	45.65	5%	43.36
		Privilege Escalation Stole Credentials	Monitor for unexpected changes to cloud users, monitor for active directory object creation and modification	Ensure user access rights; do not use domain administrator/root accounts in daily operations, network segmentation, remove unnecessary and potentially abusable authentication	Use AWS Cloudwatch and Cloudtrail, Audit, and educate users	43.36	5%	41.20
		Defense Evasion Alternate Authentication - MFA Interception	Monitor user account authentication, monitor web credentials usage from users, monitor requests of service tickets to a domain controller, monitor for third-party application logging, and login session creation	Restrict the use of authentication material outside expected contexts, configure Active Directory configuration, perform audits or scans, implement password policies, limit credential overlap across the systems, and enforce least privilege	Regular audit, user training	41.20	5%	39.14
		Discovery Container and Resource Delivery	Monitor logs for actions to gather information about containers and pods, including API calls by new or unexpected users	Limit communication with container services to secure channels, deny direct remote access through proxies, gateways, and firewalls, enforce least privilege	Policy enforcement on resource allocation	39.14	5%	37.18

(Continued)

10.3389/fcomp.2025.1557918

TABLE 12 (Continued)

Current risk score and level	Acceptable risk score and level	Critical state conducted from adversarial group	Detection	Mitigation	Prevention	Initial risk	Risk reduction	Residual risk
		Impact Network Denial of Service	Monitor network data for uncommon data flows, detect network DoS in host status, aggressive monitoring, logging, and other artifacts highlighting the health of host sensors	Filter network traffic by using services provided by Content Delivery Networks (CDN)	Create contingency plans or rerouting of network components.	37.18	5%	35.32

allow adversaries to disable recovery functions. Internal threats can also play a role in these issues. Figure 6 shows the average impact, base score, and exploitability of these tactics as **4.0**, **6.4**, and **3.1**.

5.2.5 Resource hijacking–adversarial group TeamTNT

The TeamTNT, as outlined in MITRE ATT&CK (2024c), leveraged account manipulation tactics to disrupt authentication using CVE-2023-41333. Misconfigurations, such as those associated with CVE-2024-5165, were identified during the Execution–Malicious Image phase, where attackers deployed unauthorized container images to execute malicious operations. Interdependency issues linked to CVE-2019-10200 were observed during the Discovery–Container Discovery phase, enabling attackers to exploit weak dependencies in containerized environments. We display in Figure 7 regarding impact, base score, and exploitability. It gave average values of **4.8**, **7.5**, and **2.4** from TeamTNT.

We demonstrated adversarial tactics and techniques in exploiting FIDO2 for MFA and container orchestration vulnerabilities from initial activity to impact, where existing risk assessment frameworks, such as the EBIOS risk manager, only assume external attacks while overlooking factors such as internal threats, misconfigurations, and interdependencies, which significantly amplify vulnerabilities. Quantitative metrics from the National Vulnerability Database (NVD) are used to evaluate by averaging impact, base score, and exploitability scores derived from Common Vulnerabilities and Exposures (CVE) (Booth et al., 2013), as an improvement over prior studies from Devi Priya et al. (2023), Wong et al. (2023), Mills et al. (2023), and Yosifova et al. (2021). The following subsection details how fuzzy logic harmonizes these metrics with the complexity resilience index, enabling risk level determination and mitigation strategies.

5.3 Risk level determination

Fuzzy logic, as explained in Section 4.3, is a key component of our multi-attribute risk assessment framework, implemented using the *skfuzzy* library in Python (Warner, 2022). By processing uncertain conditions with adaptive input criteria and flexible rules, fuzzy logic dynamically models imprecise and incomplete data to determine risk levels. It is a control system for managing complex processes. Table 7 shows membership functions for input and output variables for the fuzzification stage. These fuzzy inputs are then processed using rule evaluation in Table 8. We aggregate the results to produce a fuzzy output. Finally, defuzzification converts the fuzzy results into a crisp risk value. It contributes to mitigation activities in section 5.4.

5.3.1 Fuzzification

We use trapezoidal according to Section 4.3.1. Table 7 and Figure 8 outline the degree of membership functions for input and output variables for a fuzzy logic-based multi-attribute risk assessment. The variables include **Complexity resilience index**, **Impact**, **Base score**, **Exploitability**, and **Risk level**.

TABLE 13 Proposed mitigation activities: inhibit system recovery.

Current risk score and level	Acceptable risk score and level	Critical state conducted from adversarial group	Detection	Mitigation	Prevention	Initial risk	Risk reduction	Residual risk
57.45 Medium	**<= 39.0 Low**	Initial Access Valid Account - Phishing	Monitor for third-party application logging, newly constructed files from a phishing message, monitor and analyze SSL/TLS, and monitor network data for uncommon data flows	Implement audit, user training, implement network intrusion prevention, restrict web-based content, use anti-spoofing, and email authentication	Develop zero trust policy and user training	57.45	5%	54.58
		Execution User - Execution - Malicious File	Monitor for files created in unusual directories, monitor for processes spawned after opening a suspicious file	Implement behavior prevention on endpoints, implement application control, and conduct appropriate user training to bring awareness of common phishing	Implement attack surface reduction, apply user account management	54.58	5%	51.85
		Persistence Boot Autostart Execution - Registry Run Keys	Monitor executed commands and arguments that may achieve persistence by referencing it with a registry run key, monitor file modification, process creation, newly created registries, and its modification	Not Available Mitigation	Implement real-time monitoring	51.85	3%	49.26
		Privilege Escalation Modify System Process - Modify Services	Monitor for suspicious uses of the docker/podman command, such as attempts to mount the root filesystem, and monitor for newly constructed containers that repeatedly execute malicious payloads	Enforce the use of container services in rootless mode, limit access to utilities such as Docker to legitimate users only	Implement AWS CloudWatch	49.26	5%	46.79
		Defense Evasion Modify Registry	Monitor executed commands for actions that could be taken to change, conceal, and delete information; conduct remote access to network traffic flows; monitor for API calls associated with concealing the registry	Ensure proper permissions are implemented by restricting registry permissions	Audit and consider applying AWS CloudTrail	46.79	5%	44.45
		Credential Access OS Credential Dumping - LSASS Memory	Monitor commands that may attempt to access credential material, monitor for unexpected creation of memory dumps, monitor new login behavior, monitor API calls that attempt to access credentials in the process memory of the Local Security Authority Subsystem Service	Implement credential access protection, implement password policies, privileged process integrity, and user training	Apply user account management	44.45	5%	42.23
		Lateral Movement Use Alternate Authentication - Pass the Hash	Monitor requests to a domain controller, monitor newly created logins and credentials used in events, and review for discrepancies, monitor for user authentication attempts	Do not allow a domain user to be in the local administrator group, enable pass the hash mitigation to apply restrictions to local accounts, limit credential overlap across systems, apply software patches	Implement procedures that detect and alert on conditions that affect hardware and software	42.23	5%	40.12
		Exfiltration Over Web Service - Cloud Storage	Monitor executed commands that may exfiltrate data to cloud storage, monitor files for being accessed to exfiltrate data, monitor new network connections for uncommon data flows, analyze traffic patterns, and conduct packet inspection	Restrict web-based content by enforcing proxies	Monitor system health	40.12	5%	38.11

(Continued)

10.3389/fcomp.2025.1557918

TABLE 13 (Continued)

Current risk score and level	Acceptable risk score and level	Critical state conducted from adversarial group	Detection	Mitigation	Prevention	Initial risk	Risk reduction	Residual risk
		Impact Inhibit System Recovery	Monitor for unexpected deletion of cloud storage and its snapshot, monitor the registry for changes associated with system recovery, monitor the status of services involved in system recovery, monitor command-line parameters involved in inhibiting system recovery	Implement application control configured to block execution that may not be required, consider data backup, and appropriate user account management	Ensure data centers are equipped with power supplies, configure backups, and system recovery	38.11	5%	36.97

5.3.2 Fuzzy inference system

We show the flexibility of fuzzy rules through the skfuzzy library to define combinations of AND and OR operators within rules using Python's programming logical operators, according to Section 4.3.2 in Table 8, and contemplate maximum method as outlined in Section 4.3.3, reflected in Figure 9.

The following risk level and score determination is composed according to ISO 31000 (International Organization for Standardization, 2018).

- **Critical level**: Extreme chaos, scores between 90–100.
- **High level**: consider inspection and resolution, scores between 70–89.0.
- **Medium level**: Analyze after addressing high and critical risk, scores between 40–69.0.
- **Low level**: Minimal danger to intellectual property and infrastructure, scores between 0–39.0.

5.3.3 Defuzzification–risk assessment

We use a discrete centroid for defuzzification as outlined in Section 4.3.4. We are referencing the complexity resilience index, impacts, base score, and exploitability from Figures 3–7. The risk assessment results are obtained as explained in Figure 10.

Table 9 presents a multi-attribute risk assessment based on fuzzy logic, comprehensively analyzing five adversarial techniques. Data destruction and resource hijacking were identified as the highest-risk attacks among the evaluated techniques, scoring 70.47 and 74.60, respectively. These scores are categorized as "High" risk, reflecting their significant potential consequences. This underscores the urgent need for robust and proactive mitigation strategies, as these threats could cause extensive damage to critical systems without effective countermeasures. In contrast, threats such as Endpoint Denial of Service (DoS), Network DoS, and Inhibit System Recovery were classified as "Medium" risks, with scores of 59.56, 55.44, and 57.45, respectively. Although these threats can cause substantial disruptions, their lower scores suggest reduced consequences. Nevertheless, they still require attention, as their potential to degrade system performance and availability necessitates ongoing monitoring and appropriate security measures.

The following subsection outlines detailed control activities across detection, mitigation, and prevention layers to address these identified risks.

5.4 Risk mitigation and reduction activities

While traditional cloud risk assessments such as Tanimoto et al. (2014) quantify risk across asset, threat, and vulnerability dimensions, they often apply static values to mitigation efforts without accounting for the depth or layering of controls. In contrast, this study introduces a quantitative model that evaluates risk reduction based on the cumulative effectiveness of layered safeguards. The strategies in Tables 10–14 describe specific control actions arranged into three important layers: detection, mitigation, and prevention. This layered approach follows the principle from

TABLE 14 Proposed mitigation activities: resource hijacking.

Current risk score and level	Acceptable risk score and level	Critical state conducted from adversarial group	Detection	Mitigation	Prevention	Initial risk	Risk reduction	Residual risk
74.60 High	**<=69.0 Medium**	Reconnaissance Active Scanning - Vulnerability Scanning	Monitor and analyze traffic patterns and packet inspection associated with protocols that do not follow standards; monitor network data for uncommon data flows	Pre-compromised, minimizing the amount and sensitivity of data available to external parties	Implement strict access control policies	74.60	3%	72.36
		Resource Development Develop Capabilities	Use a service that may aid in tracking capabilities, analyzing malware for features associated with the adversary, and monitoring contextual data about malicious payloads	Pre-compromised	Use Amazon GuardDuty for monitoring uncommon data flows	72.36	3%	70.19
		Initial Access Staged Capabilities - External Service	Monitor anomalous external use; follow best practices for detecting adversary use for authenticating to remote services; monitor new network connections; analyze patterns and packet inspections	Disable unnecessary features, block remotely unnecessary services, limit access to remote service, use strong two-factor MFA, deny direct remote access to the internal system	Audit, User Training	70.19	5%	66.68
		Execution User Execution - Malicious Image	Monitor the local image registry, behavior of newly deployed containers, monitor attempts to take advantage of an internet-facing program, monitor the activation or invocation of an instance	Perform audits, use a trust model such as Docker Content Trust, train users to be aware of malicious images	Implement AWS CloudWatch or similar network intrusion prevention	66.68	5%	63.35
		Execution Container Administrator	Monitor suspicious command execution via AWS System Manager or Azure RunCommand, monitor process creation events in virtual machines, monitor executions of scripts within virtual machines	Limit the number of cloud accounts with permission to execute remote commands on virtual machines, and ensure these are not used for day-to-day operations	Implement user access management	63.35	5%	60.18
		Execution Deploy Container	Monitor application logs for suspicious container deployment through API management, monitor container creation to detect unknown images being deployed, monitor the start of containers/pods and their changes	Implement audits and scan images before deployment, block non-compliant ones, limit communication with container services to secure channels, and enforce least-privilege access	Apply user account management	60.18	5%	57.17
		Persistence Create or Modify System Process - System Service	Detect malicious systems using the 'systemctl' utility, audit file creation/modification, monitor new systemd services to execute repeatedly malicious payloads, analyze the content of files present on the file system	Limit software installation, restrict file and directory permissions, and implement privileged account management	Apply user account management	57.17	5%	54.31
		Defense Evasion Impair Defense	Monitor logs for API calls to disable logging, monitor changes made to cloud services, monitor executed commands that may modify components, monitor changes in firewalls, monitor changes to user account settings	Check account role permissions, use application controls, ensure proper process and file permissions to prevent adversaries from disabling logs or security services	Monitor system health, configure software, and user account management	54.31	5%	51.60

(Continued)

10.3389/fcomp.2025.1557918

TABLE 14 (Continued)

Current risk score and level	Acceptable risk score and level	Critical state conducted from adversarial group	Detection	Mitigation	Prevention	Initial risk	Risk reduction	Residual risk
		Discovery Container and Resource Discovery	Monitor logs for actions to gather information about containers, pods, including the use of discovery API calls	Limit access to resources over the network in container services to secure channels and implement network segmentation	Use AWS Lambda to automate security incidents	51.60	5%	49.02
		Lateral Movement Remote Service	Monitor interaction with network shares, new network connections, newly executed processes, and WMI objects	Conduct audits to identify potential weaknesses, prevent access to file shares, do not reuse local administrator account passwords across systems	Apply user account management	49.02	5%	46.57
		Impact Resource Hijacking	Monitor process resource usage to determine anomalous activities, monitor for common cryptomining based on executed commands, uncommon data flows, monitor new network connections, and their traffic	No Available Mitigation	Secure any access that interacts with resources from the beginning	46.57	3%	45.17

the *Certified Information Systems Security Professional (CISSP)* framework (Chapple et al., 2018), which defines residual risk as the total risk minus the effect of safeguards that are in place. To estimate how much risk is reduced, we assign effectiveness values depending on how many layers are applied. When all three layers are implemented, the effectiveness factor is set at 5%. If only two layers are used, this drops to 3%, and if only one is active, it reduces further to 1%. If no controls are used, the risk remains unchanged. These values reflect each layer's relative contribution to lowering risk.

This method aligns with the prioritization logic in the *Scaled Agile Framework (SAFe)* (Knaster and Leffingwell, 2020), specifically the Weighted Shortest Job First (WSJF) model, which includes risk reduction as a key factor when deciding which actions should be prioritized. The reduction in risk is calculated using the following Equations 21, 22:

$$\Delta R_{\text{stage}} = R_{\text{current}} \times E \tag{21}$$

$$R_{\text{after}} = R_{\text{current}} - \Delta R_{\text{stage}} \tag{22}$$

In these equations, R_{current} is the risk before applying any controls, E is the effectiveness factor based on the number of layers, and ΔR_{stage} is the amount of risk reduced. The result, R_{after}, shows the remaining risk after controls are applied. By repeating this calculation across stages, the model supports a gradual and measurable path toward acceptable risk levels.

6 Discussion

This section will discuss the comparison between the proposed multi-risk assessment with NIST Risk Management Framework (RMF) (NIST, 2012) and E-Bios Risk Manager (de la Sécurité des Systémes d'Information, 2019), and also discuss the five impacts of adversarial techniques and the limitations of the proposed multi-attribute risk assessment framework. Table 15 describes the comparisons.

1. **Data destruction:** Adversaries may delete or overwrite data to disrupt services in cloud environments by targeting snapshots and backups. Our framework quantifies the severity of data destruction, enabling prioritization of recovery efforts. Mitigation strategies include, but are not limited to, backup policies, real-time anomaly detection, and monitoring of container and file creation.
2. **Endpoint Denial of Service (DoS):** Endpoint DoS targets specific layers of the application stack, such as operating systems, servers, databases, and web applications. These attacks exploit flaws to exhaust resources or crash systems. Mitigation strategies include, but are not limited to, geo-blocking, monitoring API logs, and filtering traffic. We also monitor account events and permissions.
3. **Network Denial of Service (DoS):** Network DoS attacks flood bandwidth, paralyzing website access, email, and MFA systems. These attacks can disrupt critical container orchestration pipelines and hinder cloud-based applications. Mitigation

TABLE 15 Comparison of FuzzyFortify, NIST RMF, and EBIOS risk manager.

Aspect	FuzzyFortify	NIST RMF	EBIOS risk manager
Core objective	Quantify risk via CIA-AAN prioritization, structural complexity, and CVEs.	Control lifecycle for federal information systems.	Strategic threat identification and treatment planning.
Approach type	Modified Fuzzy AHP, Domain Mapping, Fuzzy Logic.	Control-based procedural model.	Scenario-driven, semi-quantitative.
Granularity	Fine-grained CVE-based technique scoring.	Moderate: control implementation evaluation.	High-level organizational scenarios.
Support for uncertainty	Explicitly modeled with fuzzy logic and TFNs.	Not formally addressed; deterministic.	Qualitative, via collaborative sessions.
Expert involvement	Structured pairwise judgments, aggregated.	Expert support for system categorization.	Expert-led risk workshops.
Asset complexity modeling	Yes; via Domain Mapping Matrix.	Limited; based on impact tiers.	Not structurally modeled.
Control mitigation strategy	Technique-specific mitigation per threat, layered by effectiveness (5%, 3%, 1%).	Selection from predefined control sets.	High-level strategic mitigation suggestions.
Tool support/automation	Web-based simulation.	eMASS, automation suites.	MEHARI, EBIOS-compatible tools.
Use case orientation	Technical focus on FIDO2, Kubernetes, Docker, and CVE-driven DevSecOps simulation.	Lifecycle control compliance in U.S. federal IT.	Enterprise-level risk governance (EU/regulatory).
Fills gaps in literature	Yes; combines residual risk modeling (CISSP), agile prioritization (WSJF), and multi-layer control effectiveness.	No; focuses on static control lifecycles.	No; lacks quantitative thresholds and adaptive control layering.

strategies include, but are not limited to, developing network security policies, monitoring user authentication and unusual data flow, filtering traffic, and maintaining system availability.

4. **Inhibit system recovery:** Adversaries may disable recovery tools, delete backups, or erase version histories. These include volume shadow copies and automated repair features. Mitigation strategies include but are not limited to a zero-trust policy, monitoring for suspicious use of docker commands, executing commands that may exfiltrate data to cloud storage, and redundancy protocols.
5. **Resource hijacking:** Adversaries may exploit compromised systems that use many resources to conduct cryptocurrency mining to degrade performance. The framework prioritizes Resource Hijacking as the highest risk. It informs targeted mitigation strategies, including but not limited to enhanced monitoring and analysis of traffic patterns and packet inspection and logs for suspicious container deployments that use excessive resource usage to determine anomalous activities.

Although the proposed multi-attribute risk assessment framework offers practical enhancements over traditional models, several limitations remain. Firstly, the current expert judgment model assumes equal weighting among all experts, without accounting for differences in professional experience, specialization, or confidence levels. While this simplifies aggregation and aligns with Chang's fuzzy AHP methodology, it may overlook nuances in expert credibility that could refine decision outcomes. Incorporating expert weighting using the Delphi technique in future iterations may improve the reliability of aggregated judgments.

Secondly, the domain mapping matrix lacks granularity for detailed analysis, which currently uses a binary representation (1 for correlation, 0 for no correlation) to quantify the relationship between assets and best practices. Given the evolving sophistication of cybersecurity requirements, we plan to enhance the web-based tool for more comprehensive numerical representations.

Thirdly, the framework's risk reduction output depends on the inclusion of detection, mitigation, and prevention activities. While assigning fixed effectiveness weights (5%, 3%, and 1%), inspired by CISSP's layered defense principle and the Scaled Agile Framework's prioritization logic, offers a structured basis for quantifying control impact, it assumes linear and independent contributions from each layer. This simplification may not reflect the interdependencies between controls in dynamic threat environments, especially when novel adversarial tactics bypass known defenses, resulting in a potential 0% reduction. To address this, future work should consider applying fuzzy scoring to represent control effectiveness, allowing for gradual transitions, uncertainty, and overlapping impacts among detection, mitigation, and prevention activities. Treating the framework as an adaptive system through threat intelligence (e.g., MITRE ATT&CK) and feedback from practitioners would enhance its responsiveness.

7 Conclusion and future work

We present a multi-attribute risk assessment framework through a three-step approach to address critical challenges in securing FIDO2-enabled Multi-Factor Authentication (MFA) and AWS-labeled container orchestration in cloud environments. To the best of our knowledge, no prior work has explored this specific integration. First, the framework introduces a Complexity Resilience Index, which combines objective expert judgments from a modified Fuzzy AHP process to prioritize CIA and AAN security properties, alongside a domain mapping matrix to quantify system complexity across components, interfaces, and architecture. This mapping aligns security properties with three structural levels: availability and non-repudiation at the component level, integrity, authentication, and authorization at the interface level, and confidentiality at the architectural level. Second, fuzzy logic integrates the Complexity Resilience Index with CVE metrics: impact, base score, and exploitability, enabling risk

prioritization under uncertainty. Third, the entire framework is deployed as an interactive, publicly available web-based tool to support practitioner adoption. The implementation source code is shared via GitHub, as referenced in the Availability of Source Code section.

Unlike existing frameworks, such as the EBIOS Risk Manager, which often rely on subjective and approximate assessments, our framework directly addresses the complexity inherent in cloud-native systems by aligning asset provisioning with domain best practices. It dynamically maps structural and adversarial threat metrics to help prioritize critical threats, such as resource hijacking and data destruction, thereby delivering evidence-based decisions for more targeted mitigation. This empowers organizations to respond proactively to evolving risks while considering often-overlooked vulnerabilities such as internal threats, configuration errors, and architectural interdependencies. Through adaptive input criteria and flexible rule-based inference, the framework enhances cybersecurity posture by guiding mitigation strategies that remain effective across detection, mitigation, and prevention layers. It also leverages MITRE ATT&CK intelligence to ensure that control decisions remain relevant to real-world adversarial tactics.

Looking ahead, we plan to expand the framework with a cost-benefit analysis module to quantify the operational costs and benefits of asset provision and mitigation actions. This will help define defensible mitigation timelines and support resource allocation based on cost-efficiency. We also intend to conduct testbed-based evaluations and run cybersecurity training programs using the web-based tool, allowing practitioners to validate the model's effectiveness in realistic scenarios. These efforts aim to ensure both theoretical soundness and practical applicability for securing cloud-native infrastructures.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: https://github.com/udahafeez7/public_cybersecurityexercise.git.

Author contributions

MHH: Writing – original draft, Writing – review & editing. MDH: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Supervision, Validation, Visualization, Writing – review & editing. YT: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – review & editing. YK: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This study was funded by the ICSCoE Core Human Resources Development Program Japan.

Acknowledgments

This work is supported by the Laboratory of Cyber Resilience, Information Science Division, Nara Institute of Science and Technology, Japan.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Publisher's note

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fcomp.2025.1557918/full#supplementary-material

References

Alahmad, Y., Daradkeh, T., and Agarwal, A. (2019). "Optimized availability-aware component scheduler for applications in container-based cloud," in *2019 Sixth International Conference on Software Defined Systems (SDS)*, 194–199. doi: 10.1109/SDS.2019.8768654

Alali, M., Almogren, A., Hassan, M. M., Rassan, I. A., and Bhuiyan, M. Z. A. (2018). Improving risk assessment model of cyber security using fuzzy logic inference system. *Comput. Secur.* 74, 323–339. doi: 10.1016/j.cose.2017.09.011

Assumpção, P., Oliveira, C., Ortiz, P., Melo, W., and Carmo, L. (2022). "A secure cloud-based architecture for monitoring cyber-physical critical infrastructures," in *2022 6th Cyber Security in Networking Conference (CSNet)*, 1–7. doi: 10.1109/CSNet56116.2022.9955607

Bánáti, A., Kail, E., Karóczkai, K., and Kozlovszky, M. (2018). "Authentication and authorization orchestrator for microservice-based software architectures," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1180–1184. doi: 10.23919/MIPRO.2018.8400214

Bhol, S. G., Mohanty, J., and Pattnaik, P. K. (2023). Taxonomy of cyber security metrics to measure strength of cyber security. *Mater. Today* 80, 2274–2279. doi: 10.1016/j.matpr.2021.06.228

Blaise, A., and Rebecchi, F. (2022). "Stay at the helm: secure kubernetes deployments via graph generation and attack reconstruction," in *2022 IEEE 15th International Conference on Cloud Computing (CLOUD)*, 59–69. doi: 10.1109/CLOUD55607.2022.00022

Booth, H., Rike, D., and Witte, G. A. (2013). *The National Vulnerability Database (NVD): Overview*.

Bracke, V., Santos, J., Wauters, T., De Turck, F., and Volckaert, B. (2024). A multiobjective metaheuristic-based container consolidation model for cloud application performance improvement. *J. Netw. Syst. Manag.* 32:61. doi: 10.1007/s10922-024-09835-7

Business Insight (2023). *Containers as a service market size, share | 2024 to 2031*. Available online at: https://www.businessresearchinsights.com/market-reports/containers-as-a-service-market-106392 (Accessed October 23, 2024).

Cao, C., Blaise, A., Verwer, S., and Rebecchi, F. (2022). "Learning state machines to monitor and detect anomalies on a kubernetes cluster," in *Proceedings of the 17th International Conference on Availability, Reliability and Security, ARES '22* (New York, NY, USA: Association for Computing Machinery). doi: 10.1145/3538969.3543810

Casola, V., De Benedictis, A., Mazzocca, C., and Orbinato, V. (2024). Secure software development and testing: a model-based methodology. *Comput. Secur.* 137:103639. doi: 10.1016/j.cose.2023.103639

Chang, D.-Y. (1996). Applications of the extent analysis method on fuzzy ahp. *Eur. J. Oper. Res.* 95, 649–655. doi: 10.1016/0377-2217(95)00300-2

Chapple, M., Stewart, J. M., and Gibson, D. (2018). *(ISC)2 CISSP Certified Information Systems Security Professional Official Study Guide*. SYBEX Inc., USA, 8th edition.

de la Sécurité des Systémes d'Information, A. N. (2019). *La méthode ebios risk manager—le guide*. Technical Report ANSSI-PA-048-EN, Agence Nationale de la Sécurité des Systémes d'Information, Paris, France.

Derhab, A., Belaoued, M., Guerroumi, M., and Khan, F. A. (2020). Two-factor mutual authentication offloading for mobile cloud computing. *IEEE Access* 8, 28956–28969. doi: 10.1109/ACCESS.2020.2971024

Devi Priya, V. S., Sethuraman, S. C., and Khan, M. K. (2023). Container security: precaution levels, mitigation strategies, and research perspectives. *Comput. Secur.* 135:103490. doi: 10.1016/j.cose.2023.103490

Dissanayaka, A. M., Mengel, S., Gittner, L., and Khan, H. (2020). "Vulnerability prioritization, root cause analysis, and mitigation of secure data analytic framework implemented with Mongodb on singularity linux containers," in *Proceedings of the 2020 4th International Conference on Compute and Data Analysis*, 58–66. doi: 10.1145/3388142.3388168

Emrouznejad, A., and Ho, W. (2017). *Fuzzy analytic hierarchy process*. doi: 10.1201/9781315369884

FIDO (2022). *Fast identity online*. Available online at: https://fidoalliance.org/wp-content/uploads/2022/03/FIDO-White-Paper-Choosing-FIDO-Authenticators-for-Enterprise-Use-Cases-RD10-2022.03.01.pdf (accessed November 18, 2024).

Flavia, B. J., and Chelliah, B. J. (2023). Artificial lizard search optimized fuzzy logic approach to addressing authentication and data security challenges in p2p cloud environments. *Comput. Secur.* 135:103475. doi: 10.1016/j.cose.2023.103475

Forman, E., and Peniwati, K. (1998). Aggregating individual judgments and priorities with the analytic hierarchy process. *Eur. J. Oper. Res.* 108, 165–169. doi: 10.1016/S0377-2217(97)00244-0

Gao, X., Gu, Z., Li, Z., Jamjoom, H., and Wang, C. (2019). "Houdini's escape: breaking the resource rein of linux control groups," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 1073–1086. doi: 10.1145/3319535.3354227

Geramian, A., Abraham, A., and Ahmadi Nozari, M. (2019). Fuzzy logic-based FMEA robust design: a quantitative approach for robustness against groupthink in group/team decision-making. *Int. J. Prod. Res.* 57, 1331–1344. doi: 10.1080/00207543.2018.1471236

Ghorbani Lyastani, S., Schilling, M., Neumayr, M., Backes, M., and Bugiel, S. (2020). "Is fido2 the kingslayer of user authentication? A comparative usability study of fido2 passwordless authentication," in *2020 IEEE Symposium on Security and Privacy (SP)*, 268–285. doi: 10.1109/SP40000.2020.00047

Grimes, R. A. (2020). *Hacking Multifactor Authentication*. New York: John Wiley Sons. doi: 10.1002/9781119672357

Haripriya, A. P., and Kulothungan, K. (2019). Secure-MQTT: an efficient fuzzy logic-based approach to detect dos attack in MQTT protocol for internet of things. *EURASIP J. Wirel. Commun. Netw.* 2019:90. doi: 10.1186/s13638-019-1402-8

Henricks, A., and Kettani, H. (2020). "On data protection using multi-factor authentication," in *Proceedings of the 2019 International Conference on Information System and System Management, ISSM 2019* (New York, NY, USA: Association for Computing Machinery), 1–4. doi: 10.1145/3394788.3394789

Hersyah, M. H., Hossain, M. D., Taenaka, Y., and Kadobayashi, Y. (2023). "A risk assessment study: encircling docker container assets on IAAS cloud computing topology," in *2023 6th Conference on Cloud and Internet of Things (CIoT)*, 225–230. doi: 10.1109/CIoT57267.2023.10084910

Huang, H., Sun, B., and Hu, L. (2024). A task offloading approach based on risk assessment to mitigate edge DDOS attacks. *Comput. Secur.* 140:103789. doi: 10.1016/j.cose.2024.103789

International Organization for Standardization (2018). *Risk Management—Guidelines*. ISO, Geneva, Switzerland: ISO 31000, 2018.

Jun, Z. (2017). A security architecture for cloud computing alliance. *Recent Adv. Electr. Electr. Eng.* 10, 195–201. doi: 10.2174/2352096510666170601091846

Kim, H., Lee, D., and Ryou, J. (2020). "User authentication method using fido based password management for smart energy environment," in *2020 International Conference on Data Mining Workshops (ICDMW)*, 707–710. doi: 10.1109/ICDMW51313.2020.00100

Knaster, R., and Leffingwell, D. (2020). *SAFe 5.0 Distilled: Achieving Business Agility with the Scaled Agile Framework*. Boston: Addison-Wesley Professional.

Koksal, S., Catak, F. O., and Dalveren, Y. (2024). Flexible and lightweight mitigation framework for distributed denial-of-service attacks in container-based edge networks using kubernetes. *IEEE Access* 12, 172980–172991. doi: 10.1109/ACCESS.2024.3501192

Kudo, R., Kitahara, H., Gajananan, K., and Watanabe, Y. (2021). "Integrity protection for kubernetes resource based on digital signature," in *2021 IEEE 14th International Conference on Cloud Computing (CLOUD)*, 288–296. doi: 10.1109/CLOUD53861.2021.00042

Kure, H. I., Islam, S., Ghazanfar, M., Raza, A., and Pasha, M. (2022). Asset criticality and risk prediction for an effective cybersecurity risk management of cyber-physical system. *Neural Comput. Applic.* 34, 493–514. doi: 10.1007/s00521-021-06400-0

Lee, J.-B., Yoo, T.-H., Lee, E.-H., Hwang, B.-H., Ahn, S.-W., and Cho, C.-H. (2021). High-performance software load balancer for cloud-native architecture. *IEEE Access* 9, 123704–123716. doi: 10.1109/ACCESS.2021.3108801

Mahavaishnavi, V., Saminathan, R., and Prithviraj, R. (2024). Secure container orchestration: a framework for detecting and mitigating orchestrator-level vulnerabilities. *Multimed. Tools Appl.* 84, 18351–18371. doi: 10.1007/s11042-024-19613-x

Maurer, M., and Lindemann, U. (2008). "The application of the multiple-domain matrix: Considering multiple domains and dependency types in complex product design," in *2008 IEEE International Conference on Systems, Man and Cybernetics*, 2487–2493. doi: 10.1109/ICSMC.2008.4811669

McCabe, T. (1976). A complexity measure. *IEEE Trans. Softw. Eng.* SE-2, 308–320. doi: 10.1109/TSE.1976.233837

Mills, A., White, J., and Legg, P. (2023). Longitudinal risk-based security assessment of docker software container images. *Comput. Secur.* 135:103478. doi: 10.1016/j.cose.2023.103478

Minna, F., Blaise, A., Rebecchi, F., Chandrasekaran, B., and Massacci, F. (2021). Understanding the security implications of kubernetes networking. *IEEE Secur. Priv.* 19, 46–56. doi: 10.1109/MSEC.2021.3094726

MITRE ATT&CK. (2024a). *Apt28 (sandworm team) - russian cyber espionage group*. Available online at: https://attack.mitre.org/groups/G0034/ (accessed May 18, 2024).

MITRE ATT&CK. (2024b). *Apt38 - north korean cyber threat group*. Available online at: https://attack.mitre.org/groups/G0082/ (accessed November 18, 2024).

MITRE ATT&CK. (2024c). *Teamtnt - cloud-focused cyber threat group*. Available online at: https://attack.mitre.org/groups/G0139/ (accessed May 18, 2024).

MITRE ATT&CK. (2024d). *Wizardspider - cybercrime group focused on financial operations*. Available online at: https://attack.mitre.org/groups/G0102/ (accessed May 18, 2024).

MITRE Corporation (2024). *MITRE ATT&CK Framework for Containers*. Available online at: https://attack.mitre.org/matrices/enterprise/containers/ (accessed September 22, 2024).

Mostajeran, E., Mydin, M. N. M., Khalid, M. F., Ismail, B. I., Kandan, R., and Hoe, O. H. (2017). "Quantitative risk assessment of container based cloud platform," in *2017 IEEE Conference on Application, Information and Network Security (AINS)*, 19–24. doi: 10.1109/AINS.2017.8270418

Nguyen, C. D. (2020). *A Design Analysis of Cloud-Based Microservices Architecture at Netflix*.

NIST (2012). *NIST SP 800-37 Revision 1 Guide for Applying the Risk Management Framework to Federal Information Systems*. Scotts Valley, CA: CreateSpace.

Ogundoyin, S. O., and Kamil, I. A. (2020). A fuzzy-ahp based prioritization of trust criteria in fog computing services. *Appl. Soft Comput.* 97:106789. doi: 10.1016/j.asoc.2020.106789

Outkin, A. V., Schulz, P. V., Schulz, T., Tarman, T. D., and Pinar, A. (2023). Defender policy evaluation and resource allocation with MITRE ATT&CK evaluations data. *IEEE Trans. Depend. Secure Comput*. 20, 1909–1926. doi: 10.1109/TDSC.2022.3165624

OWASP (2024a). *Cloud architecture security cheat sheet*. Available online at: https://cheatsheetseries.owasp.org/cheatsheets/Secure_Cloud_Architecture_Cheat_Sheet.html (accessed May 20, 2024).

OWASP (2024b). *Docker security cheat sheet*. Available online at: https://cheatsheetseries.owasp.org/cheatsheets/Docker_Security_Cheat_Sheet.html (accessed May 20, 2024).

OWASP (2024c). *Kubernetes security cheat sheet*. Available online at: https://cheatsheetseries.owasp.org/cheatsheets/Docker_Security_Cheat_Sheet.html (accessed May 20, 2024).

Pöhn, D., Gruschka, N., Ziegler, L., and Büttner, A. (2023). A framework for analyzing authentication risks in account networks. *Comput. Secur*. 135:103515. doi: 10.1016/j.cose.2023.103515

Renaud, K., Warkentin, M., Pogrebna, G., and van der Schyff, K. (2024). Vista: an inclusive insider threat taxonomy, with mitigation strategies. *Inf. Manag*. 61:103877. doi: 10.1016/j.im.2023.103877

Schiavone, E., Ceccarelli, A., and Bondavalli, A. (2016). "Continuous authentication and non-repudiation for the security of critical systems," in *2016 IEEE 35th Symposium on Reliable Distributed Systems (SRDS)*, 207–208. doi: 10.1109/SRDS.2016.033

Seifermann, S., Heinrich, R., and Reussner, R. (2019). "Data-driven software architecture for analyzing confidentiality," in *2019 IEEE International Conference on Software Architecture (ICSA)*, 1–10. doi: 10.1109/ICSA.2019.00009

Sheard, S. A., and Mostashari, A. (2009). Principles of complex systems for systems engineering. *Syst. Eng*. 12, 295–311. doi: 10.1002/sys.20124

Sinha, K., et al. (2014). *Structural complexity and its implications for design of cyber-physical systems*. PhD thesis, Massachusetts Institute of Technology.

Sinha, K., and Suh, E. S. (2018). Pareto-optimization of complex system architecture for structural complexity and modularity. *Res. Eng. Design* 29, 123–141. doi: 10.1007/s00163-017-0260-9

Sultan, S., Ahmad, I., and Dimitriou, T. (2019). Container security: issues, challenges, and the road ahead. *IEEE Access* 7, 52976–52996. doi: 10.1109/ACCESS.2019.2911732

Taleby Ahvanooey, M., Zhu, M. X., Ou, S., Dana Mazraeh, H., Mazurczyk, W., Choo, K.-K. R., et al. (2023). Afpr-am: a novel fuzzy-AHP based privacy risk assessment model for strategic information management of social media platforms. *Comput. Secur*. 130:103263. doi: 10.1016/j.cose.2023.103263

Tanimoto, S., Sato, R., Kato, K., Iwashita, M., Seki, Y., Sato, H., et al. (2014). "A study of risk assessment quantification in cloud computing," in *2014 17th International Conference on Network-Based Information Systems*, 426–431. doi: 10.1109/NBiS.2014.11

Tran, T. N. T., Felfernig, A., and Le, V. M. (2024). An overview of consensus models for group decision-making and group recommender systems. *User Model. User-Adapt. Interact*. 34, 489–547. doi: 10.1007/s11257-023-09380-z

Truyen, E., Kratzke, N., Van Landuyt, D., Lagaisse, B., and Joosen, W. (2020). Managing feature compatibility in kubernetes: vendor comparison and analysis. *IEEE Access* 8, 228420–228439. doi: 10.1109/ACCESS.2020.3045768

Warner, J. S. (2022). *scikit-fuzzy: Fuzzy logic toolbox for python*. Available online at: https://pypi.org/project/scikit-fuzzy/ (accessed November 02, 2024).

Wong, A. Y., Chekole, E. G., Ochoa, M., and Zhou, J. (2023). On the security of containers: Threat modeling, attack analysis, and mitigation strategies. *Comput. Secur*. 128:103140. doi: 10.1016/j.cose.2023.103140

Wu, H., Wu, Y., and Zhang, J. (2023). Risk assessment modeling with application in the accounting cloud-service industry. *Expert Syst. Appl*. 229:120526. doi: 10.1016/j.eswa.2023.120526

Yosifova, V., Tasheva, A., and Trifonov, R. (2021). "Predicting vulnerability type in common vulnerabilities and exposures (cve) database with machine learning classifiers," in *2021 12th National Conference with International Participation (ELECTRONICA)* (IEEE), 1–6. doi: 10.1109/ELECTRONICA52725.2021.9513723

Zadeh, L. A. (1965). Fuzzy sets. *Inf. Control* 8, 338–353. doi: 10.1016/S0019-9958(65)90241-X

Zahoor, E., Chaudhary, M., Akhtar, S., and Perrin, O. (2023). A formal approach for the identification of redundant authorization policies in kubernetes. *Comput. Secur*. 135:103473. doi: 10.1016/j.cose.2023.103473