Analysis of Information Security Issues in Balancing Multiple Independent Containers on a Single Server

Inna Rozlomii [1], Andrii Yarmilko [1] and Serhii Naumenko [1]

[1] *Bohdan Khmelnytsky National University of Cherkasy, 81, Shevchenko Blvd., Cherkasy, 18031, Ukraine*

Abstract

This article addresses the challenges arising from the widespread approach to optimizing resource utilization and ensuring scalability - the balancing of multiple independent containers on a single server - from an information security perspective. The risks associated with this approach and the potential consequences of vulnerabilities and attacks in such an environment are analyzed. Techniques and practices that can be used to mitigate these risks and ensure an adequate level of security during container balancing are discussed. These techniques include regular vulnerability detection and remediation in containers and their components, proper security system configuration, the use of automated vulnerability analysis, and container activity monitoring. Security practices such as access management, the use of secure container images, and regular security training for personnel are also examined. Mathematical models of various aspects of security issues during container balancing are presented, including models of unauthorized access to containers on a single server and configuration interaction models. Risk-based strategies for protection using mathematical optimization methods to reduce risks and ensure the resilience of the information system are considered. Risks are identified with insufficient isolation between containers, code vulnerabilities, inadequate authentication, and access control mechanisms. Emphasis is placed on the critical importance of security in ensuring the reliability and integrity of data and systems as a whole and the need for systematic resolution of these container-balancing information security issues. It is underscored that none of the possible approaches to container security during balancing is universal, and developing comprehensive security strategies is critically important. It is recognized as promising to apply methods for detecting abnormal loads, protection against internal threats, and integrating security measures into the container development lifecycle when developing more secure container balancing methods.

Keywords

independent container; server; information security; unauthorized access; confidential data; security vulnerabilities.

1. Introduction

With the development of modern information technologies and the utilization of containerization in cloud environments, an increasing number of companies and organizations are facing the need to balance independent containers on a single server [1]. Containerization is a virtualization technology that allows packaging and executing applications and their dependencies in isolated environments known as containers [2-3]. Each container contains everything required to run an application, including code, libraries, configuration files, and other resources. They enable applications to operate consistently in any container-supported environment, providing significant flexibility and portability [4].

Proceedings ITTAP'2023: 3rd International Workshop on Information Technologies: Theoretical and Applied Problems, November 22–24, 2023, Ternopil, Ukraine, Opole, Poland

EMAIL: inna-roz@ukr.net (Inna Rozlomii); a-ja@ukr.net (Andrii Yarmilko); naumenko.serhii1122@vu.cdu.edu.ua (Serhii Naumenko)
ORCID: 0000-0001-5065-9004 (Inna Rozlomii); 0000-0003-2062-2694 (Andrii Yarmilko); 0000-0002-6337-1605 (Serhii Naumenko)

In the case of deploying numerous containers on one server or a server cluster, there arises a necessity for container balancing [5-6]. This task involves distributing the workload (i.e., resources and computational capacity) among containers to ensure efficient resource utilization and maintain high availability and system resilience [7].

In general, the mentioned technology allows for the efficient utilization of computational resources, providing flexibility and scalability. However, one of the potential data security issues when multiple independent containers are present on a single server as a result of balancing is the possibility of one container affecting other containers residing on the same server [8-9]. Several possible collisions that can arise in such a situation include:

1. Resource Overallocation: If one container consumes an excessive amount of resources such as memory, CPU time, or network resources, it can lead to a reduction in available resources for other containers [10-11]. The potential consequence is decreased performance or operational failure.
2. Security Vulnerabilities: If one container has security vulnerabilities or operational complexities, it can compromise the entire server and impact other containers running on that server [12]. Inadequate isolation between containers can allow an attacker to propagate an attack to other containers [13].
3. Unauthorized Access: If access control for containers or the server is not sufficiently strengthened, one container may gain unauthorized access to the resources or data of other containers on the same server [14-15].
4. Configuration Interference: If one container influences the server's configuration or other containers, conflicts or unforeseen consequences may arise, potentially resulting in decreased performance or operational failure.

The purpose of this article is to analyze the risks associated with balancing multiple independent containers on a single server and the potential vulnerabilities and attacks that may pose challenges in implementing this approach. Additionally, various techniques and practices are discussed that can be employed to mitigate these risks and ensure an adequate level of information security. The proposed solutions are examined in the context of facilitating systematic and timely resolution of information security issues in container balancing scenarios, aiming to prevent unauthorized access, preserve data integrity, and ensure overall system reliability.

Load balancing in the context of independent containers that are not logically connected presents a unique set of challenges and opportunities. The fundamental idea is to distribute the computational workload efficiently across these disparate entities, optimizing resource utilization and ensuring that no single container is overburdened. This, in turn, contributes to enhanced system performance and responsiveness.

One key consideration is the absence of logical connections between these independent containers. In a traditional load balancing scenario, interconnected components can share information about their current workloads, facilitating a more informed distribution of tasks. However, in the case of independent containers, the challenge lies in devising mechanisms that allow for effective load distribution without the luxury of direct communication.

2. Related works

The relevance of information security issues in balancing multiple independent containers on a single server in the context of the increasing use of containerization in the IT industry places them at the forefront of scholarly analysis [16-17]. Remote consolidation of applications on a single server through containers can significantly simplify administration and resource management. However, it also opens up new opportunities for malicious attacks and security breaches [18].

The heightened attention to this issue is driving the search for innovative solutions and improvements to existing methods of securing containerized environments. Understanding the risks and threats associated with load balancing contributes to enhancing the resilience and reliability of these systems.

Many academic works focus on using containers to isolate applications on a single server and load-balancing methods between these containers. For instance, in [19], the effectiveness of Docker

containers in modern applications is examined, along with identified security issues related to their usage.

Other research concentrates on specific security issues associated with container adoption. In [20], the risks of using vulnerable container images and strategies to minimize these risks are discussed.

Some studies combine load balancing and security aspects. In [21], the relationship between load distribution and the capabilities for detecting and preventing attacks on load-balancing systems is explored.

Although there is a substantial body of work related to containers, load balancing, and security [22-23], certain aspects, including information security problems when balancing multiple independent containers on a single server, remain inadequately explored for several reasons. Firstly, there is instability in the realm of container identification and authentication, which can lead to unauthorized data access. Additionally, aspects of ensuring data confidentiality between containers that share server resources may create opportunities for data leakage. Another issue is that dynamic scaling and deployment of containers can impact information security by introducing unexpected vulnerabilities during the process. Furthermore, monitoring and auditing of container security are not always given due attention, potentially resulting in overlooked threats. Lastly, the absence of standardized security practices for container balancing complicates the development of effective security strategies in this domain.

3. Research methodology

As evidenced by practice, the use of containerization and cloud environments, along with balancing multiple independent containers on a single server, is becoming an increasingly common approach to resource optimization and scalability. However, this process introduces certain challenges and potential information security issues.

The roots of the problem are associated with the fact that when balancing containers on a single server, there are risks of compromising confidentiality, integrity, and availability of information. The increase in the number of containers operating on one server creates a conducive environment for attacks and abuses that may target independent containers or the server infrastructure.

To understand and develop an approach to analyzing information security problems when balancing independent containers on a single server, it is necessary to implement systematic and objective research methods. The research methodology provides a system of steps and analytical tools for examining the issue and determining appropriate information security measures. It helps structure the analysis process, identify threats and vulnerabilities, and develop protection strategies.

The first step in this methodology is formalizing the research object, which allows for a mathematical description of the system consisting of containers and a server. From this description, we move on to identifying potential threats and vulnerabilities that can affect the system. Subsequently, protection strategies are developed based on risk analysis and considering the identified vulnerabilities.

The formalization of the task involves creating a mathematical model that describes all components of the system, their interactions, and parameters. This model can be represented as a system of mathematical equations and inequalities that reflect the operation of containers and the server.

The mathematical model of the system can be expressed as follows:

1. Variables and Parameters:
   - C_i – the state of container i;
   - S – the state of the server;
   - T – the time interval;
   - $R_{ij}(t)$ – the state of interaction between container i and container j at time t;
   - $V_i(t)$ – the state of vulnerabilities of container i at time t.
2. Description of Functional Dependencies:
   - $R_{ij}(t)$ depends on the configuration of containers and the server, as well as external factors such as network traffic and the surrounding environment;

- $V_i(t)$ depends on the configuration of containers and the server, as well as external factors such as network traffic and the surrounding environment.

3. Formulation of Constraints and Conditions:

- $R_{ij}(t)$ must satisfy security requirements, i.e., $R_{ij} \leq$ [Maximum acceptable risk level];
- $V_i(t)$ must be minimized, i.e., $V_i(t) \leq$ [Maximum acceptable vulnerability level].

It is evident that $R_{ij}(t)$ and $V_i(t)$ are states described mathematically over time (t). The units and scale for these variables would be contingent upon the specific metrics used to quantify the state of interaction between containers (R_{ij}) and the state of vulnerabilities (V). For example, R_{ij} could be measured in terms of network latency, data transfer rates, or any other relevant performance metric. Similarly, V might be assessed based on the number or severity of vulnerabilities present in a container.

In terms of comparison with the maximum acceptable levels, the article establishes clear constraints and conditions for $R_{ij}(t)$ and $V_i(t)$. $R_{ij}(t)$ is constrained by security requirements, specifically $R_{ij} \leq$ [Maximum acceptable risk level]. This implies that the unit of measurement for R_{ij} should align with the chosen metric for risk assessment, and the scale should adhere to the defined maximum acceptable risk level.

Likewise, $V_i(t)$ is constrained by the minimization of vulnerabilities, expressed as $V_i(t) \leq$ [Maximum acceptable vulnerability level]. The units and scale for V_i would be dictated by the chosen metrics for quantifying vulnerabilities, and the scale should align with the stipulated maximum acceptable vulnerability level.

In essence, the units and scale used to measure R and V are context-specific, aligning with the chosen metrics for risk and vulnerability assessment. Comparing these measurements with the maximum acceptable levels ensures that the system's security is maintained within predefined thresholds, as outlined in the formalization of the mathematical model. This meticulous approach facilitates a robust analysis of the system and the formulation of protection strategies, contributing to the overall objective of achieving an optimal level of information security in container balancing scenarios.

Such a mathematical model allows for the analysis of the system and the establishment of parameters to achieve an optimal level of information security. Furthermore, based on this model, potential threats and vulnerabilities of the system can be identified, and protection strategies can be developed using mathematical optimization methods.

4. Vulnerabilities and configuration interference conflicts

One of the potential information security issues associated with having multiple independent containers on a single server is the possibility of one container influencing other containers that reside on the same server. This problem becomes more pronounced when these containers are being balanced. When containers are located on the same server and are load-balanced, they may share server resources such as memory, computing power, and network resources. If one container becomes compromised or is subjected to an attack, it can have a negative impact on other containers running on the same server.

Let's delve into this issue in more detail.

4.1. Risks and vulnerabilities in container balancing on a server

The seriousness of the security vulnerability threat when balancing multiple independent containers on a single server lies in the fact that if one container has such a vulnerability or complexity, it can lead to the compromise of the entire server and impact other containers running on that server.

For a better understanding of this point, the risk of compromising the server can be represented as a dependency that describes its value based on the number of vulnerable containers:

$$R_{dis} = \frac{N_{imp}}{N_{total}}, \quad (1)$$

where R_{dis} – risk of compromise, N_{imp} – number of vulnerable containers, N_{total} – the total number of containers. Equation (1) demonstrates that the more vulnerable containers are present on the server as a result of balancing, the higher the risk of server compromise.

Table 1 presents the types of vulnerabilities and their potential impact on the server and containers.

Table 1
Types of vulnerabilities and their potential impact on the server and containers

Vulnerability type	Possible impact
Insufficient isolation	Data leakage between containers; increased risk of attack impact
Code vulnerabilities	Execution of malicious code
Insufficient authentication	Unauthorized access
Insufficient access control	Unrestricted access to data

4.2. Unauthorized access

Unauthorized access is one of the serious security issues associated with balancing multiple independent containers on a single server [24-25]. If access control to the containers or the server itself is not properly enforced, it can open the possibility for one container to gain unauthorized access to resources or data belonging to other containers on the same server [26]. This is a paramount and pervasive information security concern. It represents a significant threat to the confidentiality, integrity, and availability of data within containerized environments.

In the context of balancing multiple independent containers on a single server, the potential for unauthorized access is heightened. The dynamic and distributed nature of containerized environments necessitates a meticulous examination of access control mechanisms to ensure the secure operation of each container. Unauthorized access can lead to data breaches, system disruptions, and compromise the overall security posture of the environment.

The potential consequences of insufficient access control are described in Table 2.

Table 2
Consequences of unauthorized access to a container

Consequence	Manifestations
Loss of confidentiality	Unauthorized container gains access to confidential data, compromising their confidentiality
Violation of integrity	Unauthorized container may alter or damage data belonging to other containers, compromising their integrity
Loss of availability	Unauthorized container may disrupt the operation of other containers, depriving them of availability
Propagation of attacks	An attacker who has gained unauthorized access to a container may be provided with opportunities to propagate attacks to other containers

Figure 1 provides a schematic representation of the server architecture with container balancing, showing the interaction between a vulnerable container and other containers. It illustrates the mechanism by which one vulnerable container can be exploited by an attacker to launch attacks on other containers and impact the server's operation.

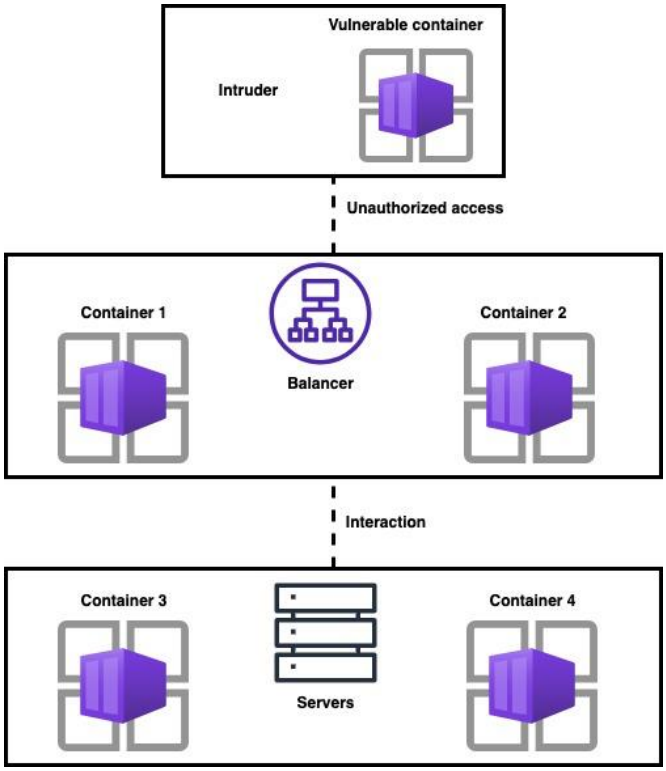

Figure 1: Interaction of an attacker with a vulnerable container and other containers in server architecture with container balancing

The mathematical model of unauthorized access to containers on a single server can be represented as an access and control level system, which includes various components and parameters. The main components of this model include:

1. Containers (C): let's define the set of all containers deployed on the server as $C = \{C_1, C_2, \dots, C_n\}$, where n – is the number of containers.
2. Server (S): the server on which the containers are deployed.
3. Users (U): the set of users who have access to the server and containers.
4. Access Rights (P): Define the set of all possible access rights as $P = \{Read, Write, Execute\}$ where $Read$ – represents the right to read, $Write$ – represents the right to write, $Execute$ – represents the right to execute.
5. Access Matrix (AM): The access matrix AM of size $(n * m)$, where n – is the number of containers and m – is the number of users, defines which users have access to which containers and with what rights. The element $AM\ [i][j]$ represents the access rights of user j to container i.
6. Authentication and Authorization System: This system defines the rules by which users authenticate and authorize for access to containers. It can be described using mathematical functions and algorithms.
7. Vulnerabilities and Attacks (V, A): The set of vulnerabilities V and possible attacks A, that attackers can use to gain unauthorized access.
8. Security Mathematical Functions: Mathematical functions can be used to determine the security level of the system, such as information entropy, attack probability, and others.

Let's express the mathematical relationships based on the outlined components in the model for unauthorized access. Mathematical security functions will be calculated according to the formula for determining information entropy:

$$H = \sum_{i=1}^{n} P_i \cdot log_2(P_i), \tag{2}$$

where P_i is the probability of a certain event (for example, unauthorized access).

This formula encapsulate the mathematical relationships within the proposed model and the information entropy provides quantitative measures for evaluating the security level of the system.

Using this mathematical model, security analysis can be performed, risks can be identified, and measures can be implemented to protect containers from unauthorized access. One of these measures is container isolation, which is based on precise parameters and restrictions. They ensure resource separation and security of container execution on a shared server. The use of specialized resource control mechanisms allows mathematical determination of resource usage limitations for each container, reducing the possibility of conflicts and resource overflows that could lead to unauthorized access and affect other containers on the server.

Let's note that in the Linux kernels, there is a mechanism called Cgroups (Control Groups), which allows limiting and controlling resources used by processes, including Docker containers. The use of Cgroups enables setting limits on resources such as the central processing unit (CPU), random-access memory (RAM), input/output (I/O), and others. Mathematically, this can be expressed as follows. Let's assume that R represents a resource (for example, CPU). Then, the limitation on resource usage by container C can be expressed as:

$$R(C) \leq R(Server), \tag{3}$$

where $R(C)$ – resource limits for container C; $R(Server)$ – available server resources.

Taking such a dependency into account allows for resource consumption limitations by one container and, thus, safeguards other containers from harm.

For modeling access levels and identifying unauthorized access possibilities, the RBAC (Role-Based Access Control) formula is used to assign roles and define access rights for each container. The access control model assesses the level of access to resources or data for each container:

$$Acces\_Level = Role\_Privileges \cap User\_Privileges, \tag{4}$$

where $Acces\_Level$ – is the access level determined as the intersection of role privileges and user privileges; $Role\_Privileges$ – are privileges assigned to a specific container role; $User\_Privileges$ – are privileges held by the user executing the container.

This formula helps identify unauthorized access possibilities when a user's access level intersects with privileges assigned to the container.

4.3. Configuration interactions

Configuration interactions are another issue associated with balancing multiple independent containers on a single server. In this scenario, the influence of one container on the server's configuration or other containers can lead to conflicts or unforeseen consequences that can significantly impact system performance and reliability.

One approach for analyzing and managing configuration interactions is to use conflict tables or dependency tables. Such tables can reflect the relationships between different configuration parameters of containers and the server, as well as define acceptable values, constraints, and recommendations for their use.

Table 3 illustrates potential conflicts between container configuration parameters and server parameters, along with provided notes on each conflict and its consequences. Such a table helps identify potential issues and avoid improper configurations that could affect system security and efficiency.

A schematic representation of the interaction of configurations between containers and the server can take various forms, depending on the system's specifics and parameters. Figure 2 illustrates the general structure of interactions between containers and the server. Such visualization helps track the interplay of configurations and identify potential issues.

In this diagram, each container and server have their configuration parameters, such as CPU, RAM, Storage, Network, Port, and Protocol. The arrows depict the interaction between containers and the server. For example, a container with a high-performance CPU interacts with a server that also has a high-performance CPU. Each container can have its configuration, which may affect interactions with other containers and the server.

Table 3
Conflict tables for analyzing configuration interactions between containers and the server

Configuration parameter	Configuration parameter influence	Server parameter	Server parameter impact	Interrelationship	Notes
CPU resource	High	CPU frame	Low	Conflict	Excessive allocation of CPU on the server can lead to resource overuse.
Number of containers	Many	Server RAM	High	Conflict	Too many containers can consume all available memory on the server.
Container OS version	Ubuntu	Server OS version	CentOS	Conflict	Different OS versions may cause compatibility issues.
Access port	8080	Server firewall	Prohibited	Conflict	A blocked port can restrict access to the container.
Network access	Allowed	Server firewall	Prohibited	Conflict	The firewall may block network access to the container.

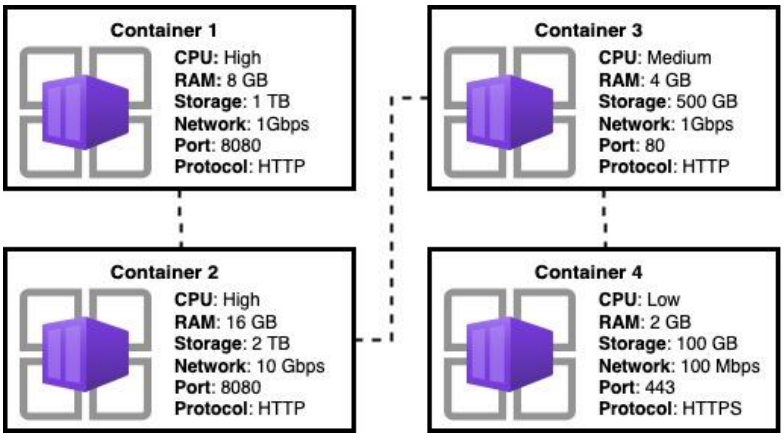

Figure 2: Interplay of configurations between containers and the server

5. Modeling the interaction of configurations

Modeling the interplay of configurations involves mathematical relationships between configuration parameters and their impact on system performance or reliability [27]. This allows us to forecast potential consequences of configuration changes and develop optimal management strategies. Here are some of the relationships:

1. Linear Interplay Model: Suppose we have two configuration parameters, A and B, and we want to determine how changes in one parameter affect the other. You can use a linear model, such as $B = k * A + b$, where k and b are coefficients of the model that define the relationship between parameters A and B. Using this formula, you can predict how changes in parameter A will impact parameter B.
2. Functional Dependency: Sometimes, the interplay of configurations can be expressed using functional dependencies. For example, if we have parameters A, B, and C, we can have a formula like $C = f(A, B)$, where f is a function that defines the relationship between parameters A and B and their influence on parameter C. This could be a mathematical function or a set of rules determining the value of parameter C based on the values of A and B.
3. Regression Models: In some cases, regression models can be used to analyze the interaction between configuration parameters and performance indicators, such as system performance or reliability. A regression model can include various factors and coefficients that determine the impact of each parameter on the performance indicator.

A scatter plot is used to illustrate the relationships between different configuration parameters, where various configuration parameters are presented on the graph [28, 29]. This helps identify correlation relationships between parameters and determine how they interact with each other.

To create a scatter plot and determine the relationships between information security parameters when balancing containers on a single server, the following parameters and their corresponding security metrics can be used, among others:

1. Parameter N – Number of containers on the server.
2. Parameter SL – System security level (numeric indicator ranging from 1 to 10).
3. Parameter SCI – Use of secure container images (binary indicator: "Yes" or "No").
4. Parameter CAM – Level of container activity monitoring (numeric indicator ranging from 1 to 5).
5. Parameter AA – Authentication and authorization level (numeric indicator ranging from 1 to 10).

These parameters represent key aspects of the information security landscape when balancing containers on a single server. The scatter plot will visualize the relationships between these parameters, allowing for the identification of correlation patterns and insights into how they interact with each other.

The proposed parameters encompass both quantitative and qualitative indicators, providing a holistic view of the system's configuration and its impact on security. For example, the binary indicator *SCI* denotes the use of secure container images, while *SL*, *CAM*, and *AA* represent numeric indicators, offering a nuanced understanding of system security levels, container activity monitoring, and authentication and authorization levels, respectively.

The impact of configuration parameters on system performance is presented in Table 4. This table provides generalized information about the impact of various configuration parameters on the performance of system components. The specific impact of each parameter may vary depending on the specific system and its requirements.

Table 4

The impact of configuration parameters on system performance

Configuration parameter	Impact on system performance
CPU power	Increasing CPU power has a positive impact on system performance, providing faster execution of computational tasks
RAM capacity	A larger amount of RAM allows the system to simultaneously process more data and programs, increasing performance.
Network throughput	High network throughput enables fast data exchange between system components, which positively affects performance
Cache memory	High network throughput enables fast data exchange between system components, which positively affects performance

6. Security strategies

After analyzing risks and identifying system vulnerabilities, developing security strategies becomes the next crucial step to ensure system safety. This phase involves devising and implementing protective measures aimed at reducing risks and ensuring system resilience. Security strategies based on risk analysis and accounting for identified vulnerabilities aim to provide effective and targeted protection for a containerized server environment. Security strategies that utilize mathematical optimization methods include:

1. Optimal Container Placement: Mathematical models and optimization algorithms determine the most efficient placement of containers on the server. This reduces potential risks and vulnerabilities while ensuring optimal resource utilization.
2. Risk Management: Mathematical models help assess risks and their impact on the system. Optimization methods identify the best approach to manage these risks, including the selection of protective measures and their priorities.
3. Vulnerability Minimization: Using mathematical methods to identify the most critical vulnerabilities in the system and developing strategies to minimize them. This includes patching vulnerabilities, enhancing security policies, and implementing other measures.
4. Resource Optimization: Utilizing mathematical models to optimize resource allocation between containers and the server while considering security aspects. This helps achieve efficient utilization of computational and network resources.

The application of mathematical optimization methods enables the development of optimal and effective security strategies, reducing risks, and enhancing system security in the context of containerized server balancing.

7. Discussions

The article brings attention to certain critical aspects that have remained inadequately explored in the existing body of literature. Specifically, the issues of instability in container identification and authentication, challenges in ensuring data confidentiality, and the impact of dynamic scaling on information security are identified as key gaps. By delving into these areas, our work contributes vital insights that complement and extend the current state of knowledge.

The absence of standardized security practices for container balancing, as highlighted in our analysis, poses a significant challenge. Our results contribute by shedding light on the intricacies of security issues specific to the dynamic environment of balancing multiple independent containers on a single server. This insight is crucial for the development of effective and tailored security strategies, filling a crucial void in the current scholarly discourse.

While some studies have explored the relationship between load distribution and security aspects, our work adds depth to this exploration. By focusing on the security problems inherent in balancing multiple independent containers, we provide a more nuanced understanding of the intersection between load balancing and security, offering valuable perspectives that go beyond the existing analyses in the field.

In conclusion, the obtained results stand out as a significant advancement in scholarly analysis by addressing critical gaps, tailoring security strategies, integrating load balancing and security considerations, and adopting a holistic approach to container security. The relevance of our findings lies in their ability to enhance the resilience and reliability of systems in the face of evolving challenges associated with the increasing use of containerization in the IT industry.

8. Conclusion

In conclusion, the article addresses critical issues related to information security when balancing independent containers on a single server. It emphasizes the need to pay proper attention to these aspects since inadequate measures can lead to serious consequences, including data compromise and threats to system security. The article proposes various techniques and practices to mitigate risks and

ensure an adequate level of security, such as regular vulnerability detection and remediation, security system configuration, the use of automated vulnerability analysis, and proper access management. Ultimately, the article underscores the importance of systematically addressing these information security issues in container-balancing environments, which are becoming increasingly prevalent. It highlights that security is critically important for ensuring the reliability and integrity of data and systems as a whole.

9. References

[1] E. Casalicchio, S. Iannucci, The state-of-the-art in container technologies: Application, orchestration and security, Concurrency and Computation: Practice and Experience 32(17) (2020) e5668.

[2] J. C. Wang, W. F. Cheng, H. C. Chen, H. L. Chien, Benefit of construct information security environment based on lightweight virtualization technology, in: 2015 International Carnahan Conference on Security Technology (ICCST), IEEE, 2015, pp. 1-4.

[3] U. Wieder, Hashing, load balancing and multiple choice, Foundations and Trends in Theoretical Computer Science 12(3–4) (2017) 275-379.

[4] P. Mahadevappa, R. K. Murugesan, Study of container-based virtualisation and threats in fog computing, in: Advances in Cyber Security: Second International Conference, ACeS 2020, Penang, Malaysia, December 8-9, 2020, Revised Selected Papers 2, Springer Singapore, 2021, pp. 535-549.

[5] R. Cziva, S. Jouet, K. J. White, D. P. Pezaros, Container-based network function virtualization for software-defined networks, in: 2015 IEEE symposium on computers and communication (ISCC), IEEE, 2015, pp. 415-420.

[6] K. Suo, Y. Zhao, W. Chen, J. Rao, An analysis and empirical study of container networks, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, IEEE, 2018, pp. 189-197.

[7] N. Singh, Y. Hamid, S. Juneja, G. Srivastava, G. Dhiman, T. R. Gadekallu, M. A. Shah, Load balancing and service discovery using Docker Swarm for microservice based big data applications, Journal of Cloud Computing 12(1) (2023) 1-9.

[8] M. Amaral, J. Polo, D. Carrera, I. Mohomed, M. Unuvar, M. Steinder, Performance evaluation of microservices architectures using containers, in: 2015 ieee 14th international symposium on network computing and applications, IEEE, 2015, pp. 27-34.

[9] R. Xie, Q. Tang, S. Qiao, H. Zhu, F. R. Yu, T. Huang, When serverless computing meets edge computing: Architecture, challenges, and open issues, IEEE Wireless Communications 28(5) (2021) 126-133.

[10] J. Watada, A. Roy, R. Kadikar, H. Pham, B. Xu, Emerging trends, techniques and open issues of containerization: A review, IEEE Access 7 (2019) 152443-152472.

[11] Z. Wang, M. Goudarzi, J. Aryal, R. Buyya, Container orchestration in edge and fog computing environments for real-time iot applications, in: Computational Intelligence and Data Analytics: Proceedings of ICCIDA 2022, Springer Nature Singapore, Singapore, 2022, pp. 1-21.

[12] Z. Li, H. Jin, D. Zou, B. Yuan, Exploring new opportunities to defeat low-rate DDoS attack in container-based cloud environment, IEEE Transactions on Parallel and Distributed Systems 31(3) (2019) 695-706.

[13] L. Xing, X. Bai, T. Li, X. Wang, K. Chen, X. Liao, S.-M. Hu, X. Han, Cracking app isolation on apple: Unauthorized cross-app resource access on MAC os~ x and ios. in: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 2015, pp. 31-43.

[14] A. R. Manu, J. K. Patel, S. Akhtar, V. K. Agrawal, K. B. S. Murthy, A study, analysis and deep dive on cloud PAAS security in terms of Docker container security, in: 2016 international conference on circuit, power and computing technologies (ICCPCT), IEEE, 2016, pp. 1-13.

[15] S. K. Mondal, R. Pan, H. D. Kabir, T. Tian, H. N. Dai, Kubernetes in IT administration and serverless computing: An empirical study and research challenges, The Journal of Supercomputing (2022) 1-51.
[16] K. German, O. Ponomareva, An Overview of Container Security in a Kubernetes Cluster, in: 2023 IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), IEEE, 2023, pp. 283-285.
[17] O. Bentaleb, A. S. Belloum, A. Sebaa, A. El-Maouhab, Containerization technologies: Taxonomies, applications and challenges, The Journal of Supercomputing 78(1) (2022) 1144-1181.
[18] M. Kaur, R. Aron, A systematic study of load balancing approaches in the fog computing environment, The Journal of supercomputing 77(8) (2021) 9202-9247.
[19] X. Gao, Z. Gu, M. Kayaalp, D. Pendarakis, H. Wang, Containerleaks: Emerging security threats of information leakages in container clouds, in: 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2017, pp. 237-248.
[20] X. Gao, B. Steenkamer, Z. Gu, M. Kayaalp, D. Pendarakis, H. Wang, A study on the security implications of information leakages in container clouds, IEEE Transactions on Dependable and Secure Computing 18(1) (2018) 174-191.
[21] X. Xie, T. Yuan, X. Zhou, X. Cheng, Research on trust model in container-based cloud service, Computers, Materials and Continua 56(2) (2018) 273-283.
[22] A. Modak, S. D. Chaudhary, P. S. Paygude, S. R. Ldate, Techniques to secure data on cloud: Docker swarm or kubernetes? in: 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), IEEE, 2018, pp. 7-12.
[23] A. Mailewa, S. Mengel, L. Gittner, H. Khan, Mechanisms and techniques to enhance the security of big data analytic framework with mongodb and Linux containers, Array 15 (2022) 100236.
[24] S. H. Han, H. K. Lee, S. T. Lee, S. J. Kim, W. J. Jang, Container image access control architecture to protect applications, IEEE Access 8 (2020)162012-162021.
[25] T. Alyas, S. Ali, H. U. Khan, A. Samad, K. Alissa, M. A. Saleem, Container Performance and Vulnerability Management for Container Security Using Docker Engine, Security and Communication Networks (2022).
[26] L. Xing, X. Bai, T. Li, X. Wang, K. Chen, X. Liao, S.-M. Hu, X. Han, Unauthorized cross-app resource access on mac os x and ios, arXiv preprint (2015) arXiv:1505.06836. URL: https://doi.org/10.48550/arXiv.1505.06836.
[27] R. Chandramouli, Z. Butcher, Building secure microservices-based applications using service-mesh architecture, Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, 2020. URL: https://doi.org/10.6028/NIST.SP.800-204A.
[28] M. Moravcik, M. Kontsek, Overview of Docker container orchestration tools, in: 2020 18th International Conference on Emerging eLearning Technologies and Applications (ICETA), IEEE, 2020, pp. 475-480.
[29] A. R. Manu, J. K. Patel, S. Akhtar, V. K. Agrawal, K. B. S. Murthy, Docker container security via heuristics-based multilateral security-conceptual and pragmatic study, in: 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), IEEE, 2016, pp. 1-14.