IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Improved Public Cloud Security With Containerization And Blockchain Technology

1.Satyajit Sudhir Nirgude
Department of Computer Engineering,
Shalaka Foundations Keystone School of Engineering, Pune, Maharashtra, India

2.Prof. Vrushali Wankhede
Department of Computer Engineering,
Shalaka Foundations Keystone School of Engineering, Pune, Maharashtra, India

Abstract:

This study focuses on creating a novel approach for improving public cloud security by adopting blockchain technology with containerization. This new methodology includes assigning each employee an individual container, isolating work environment, and logging everything that happens in the containers on an immutable ledger using blockchain. Containers facilitate workload separation in a light and scalable way, and blockchain guarantees a secure, transparent, and tamper-proof audit trail for activities, thus providing greater control and visibility in public cloud infrastructures. The approach addresses some of the critical challenges regarding unauthorized access, data breach, and compliance, thereby providing a strong framework in the context of secure cloud computing. This we do using a multitude of simulations, both at keeping integrity, security in operation, and scalability of the data. It concludes by postulating the prospects of this methodology in actual multi-industrial application and further a discussion on potential futures research directions.

Keywords:

Cloud Security, Containerization, Blockchain, Public Cloud, Immutable Logging, Hyperledger Fabric, Kubernetes, Secure Cloud Computing, Data Integrity, Distributed Ledger

1. Introduction

One of the underpinnings of the modern IT infrastructure is cloud computing: organizations now have scalable, low-cost solutions for storing and processing data. However, the establishment of public cloud service transitions bears some significant security risks, such as breaches of data, unauthorized access, and lack of visibility into the operations of the cloud. The traditional mechanisms of cloud security like firewalls and encryption are many times insufficient to keep up with the threats from new sources, particularly if these threats come across a multi-tenant public cloud environment which shares one set of resources among multiple organizations.

This paper discusses the hybrid approach to public cloud security by using two key technologies: containerization and blockchain. Containerization provides lightweight, isolated environments for applications in order to ensure that each employee's workload is securely compartmentalized. Blockchain is a decentralized and immutable form of technology with a tamper-proof ledger for recording all the activities going on in these containers. This setup forms a successful combination that enhances security and ensures transparency and traceability, thereby reducing the risks associated with the usage of public clouds.

Contributions of this work are the following:

A detailed methodology for increasing security from the cloud by using containers and blockchain.

A comprehensive discussion of technical implementation matters like container orchestration and blockchain integration.

Experimental results showcasing the security enhancement and performance overhead due to the introduced approach.

2. Background and Related Work:

The multi-tenancy nature of public cloud environments generates several security-related challenges. Issues such as data leakage and compromised mechanisms of access control are among the commonly seen issues. Lack of visibility in the activities of users also forms an important challenge in public cloud environments. Traditional partial solutions include encryption and identity management, but these cannot capture fully the dynamic behavior of cloud security threats.

2.1 Containerization in Cloud Security:

Containers now have become the all-popular way of deploying cloud applications these days with the implementation of Docker and Kubernetes. They are very lightweight and easily orchestrable. Unlike VMs, containers shared the kernel of the host operating system, thus improving resource usage and speeding up deployment times. Containers also provide very high isolation, so if a vulnerability exists in one container, it would not affect any others. Isolation is the key for securing applications in clouds against threats like privilege elevation and unauthorized access.

2.2 Blockchain for Secure Logging:

Blockchain technology received recognition due to its potential to build immutable, distributed ledgers. Blockchain was first developed for cryptocurrency and hence its core properties, including de-centralization, immutability, and transparency, enable it to be a very good candidate for secure logging in the clouds. By logging every container activity on a blockchain, organizations ensure that actions are permanently recorded and can neither be changed nor deleted for an auditable trail toward compliance and forensic analysis. There is no danger of some single point of failure, nor is there the potential for malicious insiders to manipulate the process, in decentralized approaches.

2.3 Existing Approaches:

Several research studies have indicated containers enhancing cloud security. Most, however, have been container orchestration tools, such as Kubernetes, and fewer studies on how blockchain can be integrated in the integration with the necessity of having logs that are only immutable and secured for a cloud setting. Blockchain applications in cloud security garnered most of the research interest in decentralized identity management and secure data storage. Our work extends these with containers and blockchain in order to develop a holistic security framework.

3. Proposed Methodology:

The proposed methodology includes the two critical elements: container-based isolation and blockchain-based activity logging. Both provide together the highly secured cloud environment.

3.1 Container-Based Isolation:

In this approach, each worker or cloud user is given a separate container in which he or she can work. Containers offer an isolation environment in which the risk of cross-application vulnerabilities is minimized and the impact of security breaches, if any, is also limited. For container orchestration purposes, we make use of Docker containers under Kubernetes, and this ensures hassle-free scalability and resource management.

The process of containerization includes:

Container Isolation: Each container runs an isolated instance of an application and makes sure that the acts of one user cannot affect those of others.

Resource Allocation: Containers are allocated with pre-defined or defined CPU, memory, and storage limits in order to ensure optimal performance.

Security Policies: Kubernetes offers network policies, Role-based access control (RBAC), and pod security policies with which access controls and network configurations of containers are defined.

3.2 Blockchain-Based Secure Logging:

A private, permissioned blockchain is utilized to track all activities within each container. Our choice of Hyperledger Fabric for blockchain implementation is on the basis of fine-grained access control and immutable, tamper-proof logs that can be created. Every time an event happens in the container might be the access of a file, network communication, or process creation corresponding transaction is recorded on the blockchain.

Blockchain integration involves :

Distributed Ledger: It has a decentralized ledger which operates on multiple nodes and is tamper-proof as well as loss-proof.

Immutable Logging: Everything that happens in the container is logged and, after being written to the blockchain, cannot be changed or removed.

Consensus Mechanism: Hyperledger Fabric has a permissioned consensus mechanism that permits only the allowed nodes in order to validate as well as attach transactions to the ledger.

3.3 Workflow Overview

1. Deployment of a container: The moment an employee logs-in into the cloud, it deploys a container for the session.

2. Logging of Activities: With each activity that occurs in the container-for instance, access to files or execution of a command, every activity is logged onto the blockchain

3. Audit Trail: Cloud Administrators can view the logs so they could identify suspicious activity or investigate incidents.

3.4 Security Features:

Data Integrity: With blockchain, we ensure that activity logs are not mutable and can be audited at any point for any security violations.

Access Control: We enforce role-based access control (RBAC) at both the container and blockchain layers.

Tamper Resistance:We avoid any single entity from tampering with records by storing logs on a decentralized blockchain.

4. Implementation Details:

4.1 Container Orchestration :

We used Kubernetes for container orchestration wherein we placed containers for every cloud user. The underlying reasons for using Kubernetes are as follows.

Auto-scaling: Containers are automatically scaled up and down based on user demand with efficient resource use.

Networking: Network policies enforce strict communication restrictions between containers, thus shortening the attack surface

Logging: Kubernetes integrates Fluentd for log aggregation. Container logs are forwarded to the blockchain for secure storage.

4.2 Blockchain Integration:

Hyperledger Fabric was used in developing a private permissioned blockchain network. Main constituents include:

Smart Contracts: Custom smart contracts define the logging rules for container activities.

Peers: Multiple peers validate the transactions and maintain the distributed ledger.

Ordering Service: It ensures that the transaction is ordered, and the ledgers additions are consistent.

4.3 Security Mechanisms:

Data Encryption: All data exchange between containers and the blockchain is encrypted using TLS

Integrity Verification: Hash of the container logs is kept on the blockchain for integrity.

Access Control Policies: Users were assigned roles and access was restricted based on their role.

5. Experimental Results and Evaluation:

5.1 Test Environment:

For the experiments, we created a testbed of 50 containers spread over 10 virtual machines. For the blockchain network, we used 5 peer nodes running Hyperledger Fabric.

5.2 Performance Metrics:

The performance of our solution was measured using the following parameters:

Transaction Latency: How long will it take to record activities on the blockchain.

Scalability: The ability of the system not to degrade in terms of performance as more containers are added.

Security: Ability to detect unauthorized access and ensure data integrity.

Metrics	Traditional cloud	Cloud With Containers	Cloud With Containers+Blockchain
Unauthorized Access Detection	Low	Moderate	High
Data Integrity Verification	Low	Moderate	High
Transaction Latency(ms)	N/A	50	100

6. Conclusion:

Our experiments demonstrate how the combined use of containerization and blockchain improves public clouds substantially on some aspects related to security. In turn, the logging system based on blockchain guarantees that all actions performed by containers may be logged in an immutable way, thereby enabling adequate audits and limiting the risks from insider threats. Containers also enhance workloads isolation that eliminate lateral movement within the overall cloud infrastructure. Although this solution does impose some overhead for the processing of blockchain transactions, its security trade-offs make this cost irrelevant. Further research could be in optimizing blockchain performance to further lower latency and integrate machine learning in order to detect anomalies from container logs.

Interfacing containerization and blockchain technology promises to be a robust, scalable solution for challenges of security in public cloud environments. Our methodology isolates workloads and maintains an immutable record of all activities performed. The operational security as well as the transparency are enhanced through this approach.

7.References:

1. M. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using Blockchain for Medical Data Access and Permission Management," *2016 2nd International Conference on Open and Big Data (OBD)*, Vienna, 2016, pp. 25-30.

2. K. Fan, S. Wang, Y. Ren, H. Li, and Y. Yang, "Blockchain-based Secure Time Protection Scheme in IoT," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4671-4679, June 2019.

3. Docker Inc., "What is a Container? A Standard Unit of Software," Docker, [Online]. Available: https://www.docker.com/resources/what-container. [Accessed: July 19, 2024].

4. M. J. Miorandi, M. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497-1516, 2012.

5. Hyperledger, "Hyperledger Fabric – Hyperledger," [Online]. Available: https://www.hyperledger.org/use/fabric. [Accessed: July 19, 2024].

6. A. R. Sfar, E. Natalizio, Y. Challal, and Z. Chtourou, "A roadmap for security challenges in the Internet of Things," *Digital Communications and Networks*, vol. 4, no. 2, pp. 118-137, 2018.

7. A. Bhardwaj, S. Bhardwaj, and V. Saraswat, "Data Security Concerns in Cloud Computing: A Blockchain based Solution," *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, 2019, pp. 621-625.

8. S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," [Online]. Available: https://bitcoin.org/bitcoin.pdf. [Accessed: July 19, 2024].

9. Kubernetes, "Production-Grade Container Orchestration," [Online]. Available: https://kubernetes.io. [Accessed: July 19, 2024].

10. S. Pearson, "Taking Account of Privacy when Designing Cloud Computing Services," *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, Vancouver, BC, 2009, pp. 44-52.