UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

HÉRICLES EMANUEL GOMES DA SILVA

CONFORMIDADE HABILITADA POR BLOCKCHAIN PARA ESCANEAMENTO CONFIÁVEL DE CLUSTERS KUBERNETES

CAMPINA GRANDE - PB

2023

HÉRICLES EMANUEL GOMES DA SILVA

CONFORMIDADE HABILITADA POR BLOCKCHAIN PARA ESCANEAMENTO CONFIÁVEL DE CLUSTERS KUBERNETES

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Andrey Elísio Monteiro Brito

CAMPINA GRANDE - PB

2023

HÉRICLES EMANUEL GOMES DA SILVA

CONFORMIDADE HABILITADA POR BLOCKCHAIN PARA ESCANEAMENTO CONFIÁVEL DE CLUSTERS KUBERNETES

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

Andrey Elísio Monteiro Brito
Orientador – UASC/CEEI/UFCG

Carlos Wilson Dantas de Almeida
Examinador – UASC/CEEI/UFCG

Francisco Vilar Brasileiro
Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 28 de junho de 2023.

CAMPINA GRANDE - PB

RESUMO

Este artigo apresenta um sistema de certificação de conformidade que utiliza a tecnologia blockchain e um Ambiente de Execução Confiável (TEE) para escaneamento seguro e confiável de clusters Kubernetes. O objetivo é ajudar as organizações a superarem os desafios de manter padrões da indústria e requisitos regulatórios em ambientes Kubernetes dinâmicos e distribuídos. O design do sistema incorpora um TEE para garantir processos de escaneamento seguros e aproveita a tecnologia blockchain para fornecer transparência e confiança no processo de certificação. O artigo fornece perspectivas sobre o fluxo de execução do processo de certificação de conformidade e apresenta a avaliação e resultados da implementação do conceito. Os resultados demonstram que o sistema aborda efetivamente preocupações relacionadas à confiabilidade, transparência e responsabilidade na certificação de clusters Kubernetes. No entanto, são necessários esforços adicionais de pesquisa e desenvolvimento para refinar e otimizar o sistema para implantação pronta para produção.

BLOCKCHAIN-ENABLED COMPLIANCE FOR TRUSTWORTHY KUBERNETES CLUSTER SCANNING

ABSTRACT

This paper introduces a compliance certification system that utilizes blockchain technology and a Trusted Execution Environment (TEE) for secure and trustworthy scanning of Kubernetes clusters. The aim is to help organizations overcome the challenges of maintaining industry standards and regulatory requirements in dynamic and distributed Kubernetes environments. The system design incorporates a TEE to ensure secure scanning processes and leverages blockchain technology to provide transparency and trust in the certification process. The paper provides insights into the execution flow of the compliance certification process and presents the evaluation and results of the Proof-of-Concept implementation. The findings demonstrate that the system effectively addresses concerns related to trustworthiness, transparency, and accountability in certifying Kubernetes clusters. Despite that, further research and development efforts are required to refine and optimize the system for production-ready deployment.

Blockchain-Enabled Compliance for Trustworthy Kubernetes Cluster Scanning

Héricles Silva
Universidade Federal de Campina Grande
Campina Grande, Paraíba, Brazil
hericles.silva@ccc.ufcg.edu.br

Andrey Brito
Universidade Federal de Campina Grande
Campina Grande, Paraíba, Brazil
andrey@computacao.ufcg.edu.br

ABSTRACT

This paper introduces a compliance certification system that utilizes blockchain technology and a Trusted Execution Environment (TEE) for secure and trustworthy scanning of Kubernetes clusters. The aim is to help organizations overcome the challenges of maintaining industry standards and regulatory requirements in dynamic and distributed Kubernetes environments. The system design incorporates a TEE to ensure secure scanning processes and leverages blockchain technology to provide transparency and trust in the certification process. The paper provides insights into the execution flow of the compliance certification process and presents the evaluation and results of the Proof-of-Concept implementation. The findings demonstrate that the system effectively addresses concerns related to trustworthiness, transparency, and accountability in certifying Kubernetes clusters. Despite that, further research and development efforts are required to refine and optimize the system for production-ready deployment.

Keywords

Kubernetes, cluster scanning, trusted execution environments, compliance, security, blockchain.

1. INTRODUCTION

As the use of Kubernetes continues to grow, so does the need for deep security measures to prevent potential vulnerabilities and attacks. CNCF[1] surveys highlight the finding that 96% of organizations are either using or considering using Kubernetes [1], and users are far more advanced in their adoption [2]. A key component of Kubernetes security is cluster scanning, which identifies potential issues and vulnerabilities in a cluster and ensures that recommended best practices are in place. In a recent survey, stakeholders expressed their foremost concern about misconfigurations and exposures, with 55% of the respondents highlighting this issue [3].

In this context, organizations face two pressing needs. Firstly, they require efficient mechanisms and tools that streamline the compliance process, enabling them to identify gaps and promptly address any security issues. This imperative ensures the maintenance of a robust security posture and the proactive mitigation of potential vulnerabilities. Additionally, organizations must effectively communicate their commitment to compliance with industry standards and regulatory requirements. Failure to meet these obligations can result in severe consequences, such as legal penalties, reputational damage, and a loss of customer trust.

To address these pressing needs, this paper presents the design of a compliance certification system that focuses on ensuring security, trustworthiness, and compliance within Kubernetes environments. The proposed system leverages a Trusted Execution Environment (TEE) solution to establish a secure environment for the cluster scanning process. Additionally, it incorporates blockchain technology to provide secure and immutable compliance certificates for Kubernetes clusters.

Furthermore, a Proof-of-Concept (PoC) implementation of the compliance certification system is carried out, utilizing the Marvin scanning tool[2], Intel SGX technology, and SCONE[3]. Marvin is a CLI tool designed to help Kubernetes cluster administrators ensure the security and reliability of their environments by performing extensive checks on cluster resources and identifying potential issues, misconfigurations, and vulnerabilities.

The compliance certification system outlined in this document represents a significant step forward in Kubernetes security, providing a robust and effective way to ensure the safety and compliance of Kubernetes clusters. Organizations can enjoy several advantages by implementing this system. These include improved security measures, streamlined compliance procedures, and demonstrating their commitment to industry standards and regulatory requirements. These benefits help to strengthen the organization's security posture, minimize the risk of security breaches, and increase stakeholder trust and credibility.

The rest of the paper is organized as follows. Section 2 provides an overview of popular security frameworks for Kubernetes and discusses the relevant tools upon which we implement our proposed approach. Section 3 delves into compliance certification and explains its significance within Kubernetes environments. Section 4 presents the detailed solution architecture, highlighting the key components and their interactions. The implementation details, including the integration of Marvin, Intel SGX technology, and SCONE, are discussed in Section 5. Section 6 provides an update on the current status of the implementation. In Section 7, we evaluate the compliance certification system's

[1] https://www.cncf.io/

[2] https://undistro.io/marvin/

[3] https://scontain.com/

effectiveness and present our experiments' results. Section 8 outlines future work and potential areas for improvement.

2. CLUSTER SCANNING

Cluster scanning plays a vital role in ensuring the robustness and integrity of Kubernetes clusters [4]. As organizations increasingly rely on Kubernetes for container orchestration, it becomes crucial to comprehensively evaluate these clusters' overall condition and adherence to compliance standards. This evaluation involves comprehensive cluster configuration, resources, and policies assessments to identify potential vulnerabilities, misconfigurations, and compliance gaps. By proactively scanning the cluster, organizations can detect and rectify any issues before exploitation, thus bolstering its overall security posture.

To perform these assessments, cluster scanning tools provide the necessary capabilities for in-depth examinations of Kubernetes clusters. These tools leverage various scanning techniques and algorithms to analyze different cluster aspects, including pod configurations, network policies, access controls, and adherence to best practices. Through a combination of automated checks and manual inspections, these tools generate valuable insights into the health, security, and compliance status.

By conducting regular cluster scans, organizations can mitigate risks associated with unauthorized access, data breaches, and non-compliance with industry standards and regulations. According to the State of Kubernetes 2023 report, 55% of stakeholders identified misconfigurations and exposures as their top security concerns [3]. Cluster scanning helps identify these misconfigurations, which could lead to security vulnerabilities or violations of best practices. Furthermore, the report highlights that 42% of stakeholders expressed concerns about applying policies consistently across clusters and teams. Cluster scanning aids in addressing this challenge by providing insights into policy adherence and facilitating consistent implementation.

2.1 Security Frameworks

Security frameworks are crucial in guiding and standardizing the security practices and configurations within Kubernetes clusters. These frameworks provide comprehensive guidelines, best practices, and benchmarks that organizations can follow to enhance the security posture of their clusters. By aligning with security frameworks, such as Pod Security Standards, MITRE ATT&CK, CIS Benchmark, and NSA-CISA, organizations can ensure that their clusters adhere to industry-recognized security standards and protect against common vulnerabilities and threats. These frameworks provide a valuable reference for organizations to assess, implement, and maintain robust security measures within their Kubernetes environments. In the following subsections, we will introduce several notable security frameworks that organizations can leverage to enhance the security of their Kubernetes clusters.

2.1.1 Pod Security Standards

Pod Security Standards (PSS) is a security framework for Kubernetes clusters defined and maintained by the Kubernetes community[4]. It is an open-source initiative led by the Cloud Native Computing Foundation (CNCF) and various industry experts. The PSS guidelines are continuously developed and updated in collaboration with the Kubernetes community, reflecting the latest security best practices and considerations. Pod Security Standards aim to provide a common set of security recommendations and guidelines that organizations can follow to ensure the secure deployment and operation of pods within their Kubernetes clusters. By following PSS, organizations can align their pod security practices with industry-recognized standards and enhance the overall security posture of their Kubernetes environments.

2.1.2 MITRE ATT&CK

Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) is a widely recognized and influential framework developed and maintained by MITRE Corporation[5]. As a not-for-profit organization dedicated to advancing technological innovation for the public interest, MITRE has created ATT&CK as a standardized knowledge base of adversary tactics and techniques. The framework serves as a valuable resource for the cybersecurity community, providing insights into the behaviors employed by threat actors across different platforms and environments, including containerized and cloud-native architectures like Kubernetes [5].

With its extensive adoption and referencing by security professionals, organizations, and government agencies globally, this security framework has become an industry-standard framework for understanding and combating cyber threats. It is regularly updated and expanded through collaborative efforts involving experts from diverse sectors. By adopting MITRE ATT&CK, organizations can gain a deeper understanding of potential attack vectors, enhance their threat detection capabilities, and fortify the security posture of their Kubernetes clusters. The framework offers a structured taxonomy of adversary techniques, empowering organizations to proactively identify and address security gaps, ultimately strengthening their defenses against evolving threats.

2.1.3 CIS Benchmark

The CIS (Center for Internet Security) Benchmark is a set of best practice guidelines and configuration recommendations developed by the CIS, a nonprofit organization committed to enhancing cybersecurity readiness and defense. The framework focuses on defining security configurations and hardening guidelines for various software, platforms, and systems, including Kubernetes. The CIS Benchmark for Kubernetes results from collaborative efforts by security professionals, industry experts, and CIS members who possess deep knowledge and expertise in Kubernetes security.

The framework provides detailed configuration recommendations for various components of a Kubernetes cluster, including the control plane, worker nodes, and network policies. It covers various security aspects, such as authentication, authorization, network security, logging, and auditing. By following the recommended security controls, configurations, and practices outlined in CIS Benchmark, organizations can significantly enhance the compliance of their Kubernetes clusters, aligning with industry-recognized best practices and reducing the risk of potential security breaches and unauthorized access, ensuring a strong security posture within their Kubernetes deployments.

[4] https://kubernetes.io/docs/concepts/security

[5] https://attack.mitre.org/

2.1.4 NSA-CISA

The NSA-CISA Kubernetes Hardening Guidance is a comprehensive resource jointly developed by the U.S. National Security Agency (NSA) and the Cybersecurity and Infrastructure Security Agency (CISA) [6]. This collaboration aims to provide organizations with strong recommendations and best practices for securing their Kubernetes environments.

The guidance covers critical aspects of Kubernetes security, including secure configurations, access controls, network policies, and monitoring practices. It draws upon the extensive expertise and insights of the NSA and CISA, ensuring organizations can align their Kubernetes security practices with industry-leading standards.

Adhering to the NSA-CISA Kubernetes Hardening Guidance allows organizations to enhance the resilience and integrity of their Kubernetes clusters. The recommended security measures provided in the guidance can help organizations effectively mitigate potential risks and security breaches, thereby strengthening their overall security posture in Kubernetes environments.

2.2 Undistro

Not only major state-funded agencies aim to address Kubernetes security concerns. Startups such as Getup[6], a Kubernetes-specialized company, have introduced the Undistro initiative[7], aimed at creating products that facilitate seamless management of Kubernetes environments.

The Undistro initiative offers innovative tools and products designed to address businesses' unique challenges in managing their Kubernetes clusters. By leveraging their deep understanding of Kubernetes intricacies, Getup aims to provide organizations with simplified, centralized, and efficient management solutions.

Getup's commitment to the Undistro initiative showcases its dedication to empowering organizations with the necessary tools for successful Kubernetes management. With their comprehensive approach and continuous innovation, Getup aims to simplify Kubernetes management, aligning with the evolving needs of businesses and unlocking the full potential of Kubernetes for their operations.

2.2.1 ZORA

As part of the Undistro initiative, Getup has introduced Zora, a cutting-edge tool designed to streamline and enhance Kubernetes cluster scanning. Zora is a powerful solution for organizations seeking comprehensive and automated scanning capabilities to ensure their Kubernetes environments' health, security, and compliance.

At its core, the tool follows a plugin-based architecture, allowing users to integrate scan sources encompassing different aspects of the Kubernetes cluster. These plugins act as data sources, providing the necessary information to conduct thorough scans. Zora incorporates plugins for resource configurations, policies, and best practices, covering a wide range of potential issues and vulnerabilities.

Operating on a periodic scanning model, it conducts regular scans of the Kubernetes clusters. This proactive approach ensures that any newly introduced issues or configuration changes are promptly detected and reported. By scanning periodically, organizations can stay informed about the health and compliance of their clusters, enabling them to take timely corrective actions.

The scan results generated are presented clearly and concisely, highlighting the identified issues, vulnerabilities, and misconfigurations. Zora's user-friendly interface allows users to easily navigate the scan findings, facilitating effective issue resolution and remediation.

With its architecture and periodic scanning capabilities, Zora simplifies the process of monitoring Kubernetes clusters for potential security risks, compliance gaps, and performance bottlenecks. By integrating it into their Kubernetes environments, organizations gain valuable insights into their clusters' overall health and security posture, enabling them to maintain a robust and secure infrastructure. Furthermore, Zora supports plugins, such as Popeye and Marvin, to perform specific tests and assessments, which will be discussed in the following sections.

2.2.2 Marvin

Marvin, an open-source command-line interface (CLI) tool developed as part of the UnDistro initiative, serves as a valuable asset for Kubernetes cluster administrators seeking to strengthen the security and reliability of their environments.

Marvin utilizes the power of CEL (Common Expression Language)[8] expressions to perform in-depth checks on cluster resources. These expressions enable Marvin to evaluate the state of various cluster components, configurations, and policies against predefined rules. By applying CEL expressions, Marvin can identify deviations from best practices and industry standards, allowing administrators to take proactive measures to ensure compliance and enhance security.

One notable aspect of Marvin is its integration with Zora, where it serves as a plugin. Zora incorporates Marvin's capabilities to implement checks derived from renowned security frameworks such as Pod Security Standards, MITRE ATT&CK, and NSA-CISA. This integration allows organizations to leverage the collective knowledge and guidelines provided by these frameworks to assess and bolster the security posture of their Kubernetes clusters.

The CEL expressions Marvin utilizes enable fine-grained control over the checks performed on the cluster resources. Administrators can define custom expressions or utilize built-in expressions to tailor the scanning process according to their specific requirements. This flexibility empowers administrators to prioritize security and reliability considerations that align with their organization's unique needs.

Administrators can proactively identify and address potential vulnerabilities, misconfigurations, and compliance gaps by employing Marvin as part of their Kubernetes cluster management strategy. The combination of Marvin's CEL-based checks, integration with Zora, and support for established security frameworks enhances the overall security posture of Kubernetes clusters, providing administrators with peace of mind and confidence in the reliability of their environments.

[6] https://getup.io/

[7] https://undistro.io/

[8] https://opensource.google.com/projects/cel

3. COMPLIANCE CERTIFICATION

In the fast-paced and highly regulated business environment of the present times, ensuring compliance with industry standards and regulatory requirements is no longer just a preference but a necessity. Organizations are now compelled to adhere to specific compliance standards, security guidelines, and data protection regulations imposed by governmental bodies and industry regulators. Compliance is a crucial defense mechanism against the legal consequences of cybersecurity incidents, providing evidence of due diligence and responsible business practices. In particular, maintaining compliance becomes even more complex due to these environments' dynamic and distributed nature when it comes to Kubernetes clusters. Organizations face the daunting task of ensuring their infrastructure meets the required standards while mitigating potential risks and vulnerabilities.

3.1 Kubernetes Compliance Challenges

One of the major challenges in the compliance process is the scale and complexity of modern Kubernetes deployments. Administrators often manage multiple clusters, applications, and microservices, making ensuring consistent compliance across all components increasingly difficult. Resource allocation, configuration management, patching, vulnerability assessments, and adherence to security frameworks further compound the problem. As discussed in Section 1, it is important for organizations to have efficient tools and communication to maintain a strong security posture and prevent severe consequences for any compliance breaches.

3.2 Mitigation Strategy

In light of this, a compliance certification system can provide a structured and systematic approach to evaluate and validate the adherence of Kubernetes clusters to specific standards and regulatory frameworks. It enables organizations to demonstrate their commitment to data privacy, security, and regulatory compliance to stakeholders, customers, and regulatory authorities.

By implementing a compliance certification system, organizations gain several significant benefits. Firstly, it helps identify and address compliance gaps, misconfigurations, and vulnerabilities that may exist within the cluster environment. This proactive approach allows organizations to rectify issues before they can be exploited, mitigating the potential for data breaches, unauthorized access, or regulatory violations.

Secondly, a compliance certification system provides a means to monitor and track the compliance status of Kubernetes clusters consistently. It enables organizations to ensure ongoing adherence to compliance requirements, even as the cluster environment evolves and scales over time. Regular assessments and evaluations help maintain a strong security posture, reducing non-compliance risk and associated consequences.

Moreover, a compliance certification system promotes transparency and accountability. Organizations can generate compliance reports and documentation, providing evidence of their commitment to regulatory compliance. These reports can be shared with auditors, regulatory bodies, or customers, instilling confidence in the organization's ability to protect sensitive data and meet industry-specific requirements.

3.3 Implementation Concerns

Implementing a compliance certification system introduces several concerns that must be addressed. These concerns include

I. Trustworthiness of the scanning process: The auditee needs assurance that the compliance checks are performed accurately and reliably without tampering or manipulation. The potential for malicious actors to compromise the scanning process raises concerns about the integrity of the results.
II. Adulterated compliance checks: There is a risk of unauthorized modifications or adulteration of the compliance checks, which can lead to inaccurate or false results. The certifier must ensure the scanning process remains free from external interference or unauthorized changes.
III. Lack of transparency: The lack of transparency in the scanning process can create doubts about the validity of the compliance certification. Organizations must demonstrate the scanning process's transparency and accountability to stakeholders, customers, and regulatory authorities.
IV. Need for trust in the scanning tool: Organizations require assurance that the tool used for compliance checks is reliable and trustworthy. They need to have confidence that the tool operates with integrity guarantees and is generated from open-source code that can be audited and verified, not exhibiting malicious behavior.
V. Mitigating reliance on trust: Organizations may want to minimize the need for trust in the scanning process itself. They may seek solutions that do not require blind trust in the parties involved and provide additional measures to verify the integrity and accuracy of the compliance checks.

Addressing these concerns is crucial to ensure the effectiveness and credibility of the compliance certification system. Accordingly, we can develop strategies and solutions that enhance the trust, reliability, and transparency of the scanning process.

4. SOLUTION ARCHITECTURE

In this section, we will explore the solution architecture of our compliance certification system. Our architecture addresses the challenges associated with ensuring the trustworthiness and transparency of the scanning process for Kubernetes clusters. By leveraging technologies that have recently become popular, we aim to enhance trust, integrity, and accountability in the certification process.

We will provide a high-level system architecture overview, focusing on the key components and their roles. Additionally, we will discuss the design decisions made during the development process, considering the trade-offs and challenges encountered. It will examine each component's functionalities, key features, and technical details, such as the programming languages, frameworks, and libraries used.

4.1 Overview

The solution architecture addresses the main concerns surrounding the compliance certification process for Kubernetes clusters. By incorporating a Trusted Execution Environment

(TEE) solution and leveraging blockchain technology, we provide mechanisms that directly respond to these concerns.

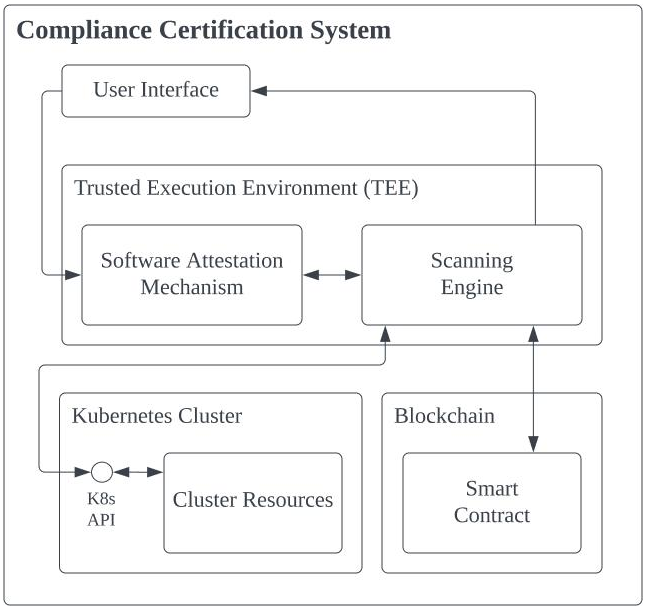
Firstly, integrating a TEE ensures the trustworthiness of the scanning process by creating a secure and isolated environment. This integration mitigates the risk of adulterated compliance checks and unauthorized tampering of critical operations. The TEE establishes a foundation of integrity, giving organizations confidence in the accuracy and reliability of the certification process [7].

Additionally, blockchain technology is crucial in addressing the lack of transparency and the need for trust in the scanning tool. Utilizing a decentralized and tamper-resistant ledger ensures the transparent recording of certification-related information that cannot be manipulated. The blockchain provides auditable proofs and establishes a verifiable chain of trust, reducing reliance on any single entity or tool [8]. This technology enhances transparency and accountability, instilling confidence in organizations and stakeholders.

The TEE and blockchain technology are strategically incorporated into our solution architecture to mitigate compliance certification concerns for Kubernetes clusters. Together, they ensure trustworthiness, transparency, and integrity in the scanning process, providing organizations with a reliable and robust solution for their compliance needs.

In summary, the solution architecture addresses the challenges of trust, transparency, and accountability, establishing a secure and auditable environment that enhances the reliability and confidence in the compliance certification process for Kubernetes clusters. *Figure 1* illustrates the solution architecture. It depicts the integration of TEE and blockchain technology in the compliance certification process for Kubernetes clusters. This diagram can help organizations better understand the technical aspects of our solution and how it addresses their compliance concerns.

Figure 1 - Compliance certification system diagram.



4.2 System Components

The compliance certification system is a comprehensive framework encompassing various essential components, working synergistically to guarantee the scanning process's accuracy, security, and auditability for Kubernetes clusters. These components collectively form a robust ecosystem that enables organizations to effectively assess and validate their clusters' compliance status. The system uses important components to give administrators and users the tools to identify issues, follow industry standards and regulations, and keep their security strong, working together to create a reliable foundation for the compliance certification system.

4.2.1 Scanning Engine

The system's core is the scanning component, which performs extensive checks on Kubernetes cluster resources and configurations. It leverages advanced scanning techniques to identify potential issues, misconfigurations, and vulnerabilities. The scanning component evaluates the clusters against industry standards and regulatory frameworks, providing valuable insights into their compliance status.

4.2.2 Trusted Execution Environment

A Trusted Execution Environment (TEE) is a hardware-based technology that offers a highly protected execution environment separate from the operating system and other software components. It ensures the integrity and confidentiality of sensitive data by encrypting and isolating it from potential threats [7]. The compliance certification system requires a Trusted Execution Environment to ensure maximum security and auditability. The TEE creates a completely secure and isolated environment for all critical operations related to the scanning process, guaranteeing sensitive data's confidentiality and integrity.

In the case of running the compliance system within the client's infrastructure, the TEE ensures the integrity and confidentiality of sensitive data by creating a highly protected execution environment separate from the operating system and other software components. This is particularly important because the client may have incentives to falsify test results or manipulate the scanning process. The TEE establishes a trust and controlled environment for scanning activities, preventing unauthorized modifications and maintaining the credibility of the compliance certification system.

On the other hand, if the compliance system is hosted in the cloud, the TEE still serves a vital purpose. It securely stores and manages the credentials required to interact with the blockchain. These credentials must be protected to prevent unauthorized access and ensure the authenticity of transactions recorded on the blockchain.

In both scenarios, the TEE within our compliance system establishes a trusted and controlled environment for scanning activities, safeguarding sensitive data, and providing a verifiable trail of all operations performed. It helps maintain the security, trustworthiness, and accountability required to meet regulatory and industry standards.

4.2.3 Attestation Mechanism

In the context of this system architecture, attestation is paramount as it verifies that the components used in the cluster scanning process are the same as the audited and approved ones and have not been compromised. By attesting the components' integrity,

users can have confidence that the scanning engine and other software that may be included in the solution implementation have not been tampered with or modified in any unauthorized way. This assurance is vital to maintain the credibility of the compliance certification process and ensures that the results obtained from the scanning activities are reliable and accurate. Attesting the components helps establish a strong foundation for trust and enhances the overall security and compliance posture of Kubernetes clusters.

4.2.4 Blockchain

Blockchain technology, specifically through smart contracts, is crucial in achieving transparency, traceability, and reliability in the compliance certification system. By leveraging smart contracts, the emission of certificates can be executed securely and automatically, ensuring accuracy and consistency. Smart contracts are self-executing agreements that are deployed onto the blockchain. They define the terms and conditions of a transaction and automatically execute the agreed-upon actions once the predetermined conditions are met [9]. In the context of compliance certification, smart contracts can be utilized to define the criteria and requirements for issuing certificates.

When a compliance assessment is completed and meets the specified criteria, a smart contract can automatically trigger the emission of a certificate. The smart contract ensures that the certificate emission process is consistent, transparent, and reliable, as it eliminates the need for manual intervention and minimizes the potential for human error or manipulation. This automation ensures that certificates are emitted promptly and accurately, enhancing the overall efficiency of the certification process.

Furthermore, the utilization of smart contracts provides traceability throughout the certification lifecycle. Every step, from the assessment to the emission of the certificate, is recorded on the blockchain, creating an immutable and auditable trail of all transactions. This traceability enables stakeholders to verify the authenticity and integrity of certificates, ensuring transparency and trust in the certification process. It also facilitates easy retrieval and auditing of certificate-related information, simplifying compliance management and regulatory reporting.

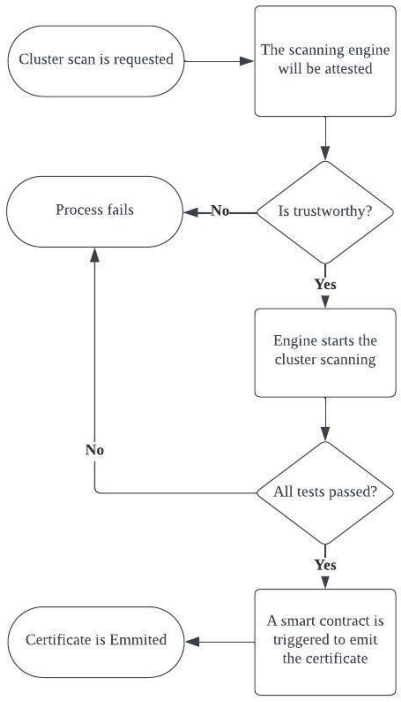
4.2.5 User Interface (UI)

The user interface component serves as the gateway for administrators and users to interact with the compliance certification system. It can take the form of a user-friendly command-line interface (CLI), allowing users to configure scanning parameters, monitor the process, and view the certification results. The CLI provides intuitive controls and displays relevant information, ensuring ease of use for administrators and users.

4.3 System Interaction

The system interaction flow diagram provides a high-level overview of the compliance certification process for Kubernetes clusters. It illustrates the key interactions between the components involved, highlighting the sequence of activities and decision points. *Figure 2* outlines the high-level flow, starting with the request for a cluster scan and ending with the certificate issuance, provided the scanning engine is trustworthy and the cluster passes all tests.

Figure 2 - High-level system interaction flow diagram.



4.4 Discussion on Design Decisions

To create an effective and dependable compliance certification system, we carefully considered several design options and trade-offs. This section covers the key design decisions and challenges faced during the process.

One of the most critical decisions made was to use a Trusted Execution Environment to carry out crucial operations within the system. After looking at some software-based and hardware-based security solutions, implementing a TEE was ultimately chosen because of its superior security guarantees, isolation capabilities, and data protection features. With a TEE, we can ensure that critical operations are performed with integrity and confidentiality, reducing the risk of unauthorized access or tampering. However, it is essential to acknowledge that this strategy comes with particular challenges and considerations. Firstly, implementing a TEE requires hardware support, which may involve additional costs for deployment and maintenance. Further, the complexity of this integration and the availability of compatible software components can pose implementation challenges that must be carefully addressed.

Despite hardware support and integration challenges, the benefits of using a TEE outweigh the drawbacks. The compliance certification system establishes a foundation of trust, integrity, and accountability by combining a TEE with a robust auditing mechanism. These design decisions ensure that critical operations are executed securely, sensitive data is protected, and transparency and auditability are provided, reinforcing the reliability of the scanning process for Kubernetes clusters and the integrity of scanning results.

Another important decision was incorporating blockchain technology and smart contracts into the compliance certification system. The blockchain is a foundational layer that ensures transparency, traceability, and reliability during certification. Although centralized databases and traditional data storage methods were considered, they lacked the desired level of immutability, transparency, and decentralized trust. With the use of blockchain technology, certification-related data is recorded in a secure and tamper-proof manner. The use of smart contracts further enhances the functionality of the blockchain by enabling automated and self-executing agreements.

However, one of the main challenges faced is the scalability of blockchain networks, especially when dealing with a large volume of certification-related transactions. The decentralized nature of blockchain requires consensus mechanisms and network participants to validate and verify each transaction, which can result in slower processing times and higher resource requirements. Furthermore, it is vital to carefully consider privacy and data protection when integrating blockchain technology and smart contracts. While blockchain provides transparency and immutability, certification-related data becomes visible to all participants in the network, raising concerns about the confidentiality of sensitive information and the need for appropriate data access controls. The contracts should prioritize protecting sensitive information, ensuring no sensitive data is exposed publicly. Addressing these challenges requires careful architectural design, optimization techniques, and ongoing research and development in blockchain technology. By understanding and mitigating these drawbacks, the compliance certification system can effectively harness the benefits of blockchain and smart contracts while minimizing potential limitations.

5. IMPLEMENTATION

In this section, we provide a comprehensive overview of the implementation of the compliance certification system. Building upon the architectural design discussed in the previous sections, we present a Proof of Concept (PoC) implementation that demonstrates the key functionalities and interactions of the system. The implementation is a tangible manifestation of the theoretical concepts, showcasing how the compliance certification system can be effectively deployed and utilized in real-world scenarios. We discuss the technologies and tools employed in the implementation process, highlighting the necessary components, configurations, and integration considerations. Furthermore, we provide insights into the challenges encountered during the implementation and the corresponding solutions devised to overcome them. The implementation details presented in this section contribute to the practical understanding and validation of the compliance certification system, providing valuable guidance for organizations seeking to deploy a robust and secure solution for Kubernetes cluster compliance.

5.1 Technical Stack

To ensure the compliance certification system works properly, we carefully selected a technical stack comprising different components that work together seamlessly. This section will review the technical stack's details and explore the technology choices made for the main components, including the scanning engine, trusted execution environment (TEE) solutions, and blockchain integration. We will highlight their functionalities, key features, and the technologies we used. By understanding the technical stack, we can make informed decisions and ensure the system is effective. This solid foundation is what makes the compliance certification system work well.

5.1.1 Scanning Engine

The scanning engine plays a crucial role in evaluating the compliance and security posture of Kubernetes clusters. In our implementation, we have chosen to leverage the power of Marvin, an open-source command-line interface (CLI) tool developed as part of the Undistro initiative. Marvin offers a group of features and capabilities for continuous monitoring and assessment of Kubernetes clusters, including built-in checks from frameworks such as PSS, NSA & CISA Kubernetes Hardening Guidance, and MITRE's ATT&CK. To enhance its scanning capabilities, Marvin can be integrated with Zora, allowing the configuration of periodic scans and access to an intuitive and unified interface with detailed information about detected issues and affected resources through the Zora Dashboard. This integrated approach comprehensively evaluates Kubernetes clusters against industry standards, regulatory frameworks, and best practices.

5.1.2 Trusted Execution Environment

The Trusted Execution Environment solution in place for the compliance certification system implementation incorporates the powerful technologies of Intel SGX and SCONE Platform for Confidential Computing, ensuring security and integrity.

Intel SGX is a hardware-based security technology that provides secure enclaves and isolated execution environments within the CPU, where sensitive code and data can be protected from unauthorized access. It guarantees the confidentiality and integrity of critical operations and data by encrypting and isolating them from potential threats [10].

SCONE complements Intel SGX by providing a framework for deploying secure Linux containers within the enclaves. It enables the execution of containerized applications while ensuring the confidentiality and integrity of their code and data. SCONE offers seamless integration with Intel SGX, simplifying the deployment and management of secure containers within the TEE.

The combination of Intel SGX and SCONE provides a robust foundation for the compliance certification system's security and isolation requirements. This ensures that critical operations, such as the scanning engine and certificate emission process, are executed within a trusted environment, safeguarding sensitive data from potential threats.

5.1.3 Attestation Mechanism

Within the TEE solution implemented using SGX and SCONE, the SCONE Configuration and Attestation Service (CAS) is vital in providing attestation capabilities. The remote attestation process enables a challenger application to gain confidence that an enclave is genuinely running on a machine with Intel SGX enabled [10]. Through this process, the application obtains crucial security properties about the machine. At the end of the attestation process, the challenger can verify the identities of the enclaves: the enclave's identity (MRENCLAVE) and the signer's identity (MRSIGNER). The MRENCLAVE results from a hash operation using the SHA-256 algorithm, involving a record of all the operations performed during the enclave's creation process and all

content related to the enclave's memory page, including code and page security flags.

Since the code is included in the hash operation, any modifications or tampering attempts on the code will result in a different MRENCLAVE value. As a result, comparing the hash verifies that the audited code matches the code running within the enclave. This mechanism guarantees that the audited code is indeed the code to be executed, eliminating concerns about unauthorized modifications or substitutions.

5.1.4 Blockchain

The Ethereum blockchain has been selected as the preferred blockchain component for the Proof-of-Concept implementation. This widely adopted platform offers a robust infrastructure for building decentralized applications and executing smart contracts.

By integrating the Ethereum Blockchain, the compliance certification system benefits from the inherent features of blockchain technology, including transparency, immutability, and decentralization. The Ethereum network serves as a trusted and tamper-resistant ledger, ensuring the integrity and transparency of certification-related transactions.

The compliance certification system utilizes Ganache as a local Ethereum blockchain for testing and development, providing a controlled environment to simulate Ethereum network behavior. It allows developers to simulate various scenarios, such as network congestion and contract interactions, providing valuable insights into the behavior and performance of the compliance certification system. As the primary programming language for smart contracts, Solidity benefits from its extensive documentation, a large developer community, and a wide range of libraries and frameworks that streamline the development process, enabling the creation of secure and efficient smart contracts on the Ethereum platform. Smart contracts, written in Solidity, enable the system to automate the certification process with predefined rules and conditions. These self-executing contracts automatically trigger the emission of certificates when specific criteria are met, eliminating the need for manual intervention and reducing the risk of human error or manipulation [9].

The decision to use the Ethereum Blockchain and Solidity for smart contracts is driven by the platform's wide adoption, robustness, and extensive developer community. Ethereum's rich ecosystem and tooling support facilitate developing and deploying secure and efficient smart contracts.

5.1.5 User Interface

A Command-Line Interface (CLI) makes interacting with the compliance system easier. It utilizes text-based input and output, which is familiar and accessible to users with varying levels of technical expertise. The CLI initiates the scanning process and oversees the compliance certification workflow, interacting with the Marvin scanning engine responsible for conducting scans of the Kubernetes clusters. Users can initiate scanning by invoking Marvin through the CLI and passing relevant parameters and configurations. This integration guarantees seamless communication between the user interface and the scanning engine.

Upon completion of the scanning process, the CLI receives the test results from Marvin. It analyzes the results to determine whether all the tests have passed or whether any issues or vulnerabilities have been identified. If all tests have successfully passed, it triggers the execution of a smart contract on the Ethereum blockchain.

The CLI is built with Golang to ensure a lightweight and reliable design. Golang's simplicity and performance can seamlessly interact with the scanning engine through the Marvin binary, initiating the scanning process and retrieving test results. The utilized package *go-ethereum* facilitates interaction with the smart contract on the Ethereum blockchain. This package provides a robust set of features for managing blockchain transactions, querying contract states, and handling cryptographic operations required for secure communication with the smart contract. With Golang, the code is efficient, easy to read, and has minimal dependencies, which enhances security and ease of maintenance.

Aside from enabling user interaction, it is crucial to note that the user interface component must function within the Trusted Execution Environment. Like the scanning engine, the user interface component should undergo a rigorous auditing and attestation process to ensure its trustworthiness and reliability. Auditing involves scrutinizing the CLI code, configuration, and access controls to expose potential vulnerabilities that malicious actors could exploit. Through attestation, the integrity and authenticity of the user interface are verified, guaranteeing it has not been tampered with or compromised.

5.2 Trusted Execution Mechanisms

Some security mechanisms are extremely important in safeguarding sensitive information within the compliance certification system, such as blockchain wallet private keys, the Kubernetes server address, and tokens. To achieve this, the system incorporates SCONE CAS as both the central point for managing configurations and secrets and as a key component of the attestation architecture. It allows multiple applications to be registered and provides an API for defining security policies and managing secrets. Applications are registered in SCONE CAS through YAML files, which specify the expected enclave identity and define secrets and configurations. The sequence of instances of security policies is called a session. The security policies enforced by SCONE CAS ensure the secure storage of sensitive information and control access to it, enhancing the overall security and integrity of the compliance certification system.

The structure of a SCONE session YAML file, as illustrated in *Figure 3*, includes some sections that define the session's properties and configurations. The session name is provided at the beginning and serves as a unique identifier for the session.

Figure 3: SCONE policy structure.

```
name: <session_name>
version: "0.3"

services:
  name: <service_name>
  attestation:
    mrenclave: [$MRENCLAVE]
  command: ./compliance scan
  pwd: /
  environment:
    SECRET: $$SCONE::secret_name$$
```

```
13|  secrets:
14|      - name: secret_name
15|        kind: ascii
16|        value: <secret_value>
```

The "services" section (lines 4-11) defines the services that will be included in the session. Each service is assigned a name and can have specific attestation requirements to ensure its integrity. A service configuration consists of the command to be executed, any environment variables, and the process working directory from which the path resolution of relative paths starts. The attestation section allows for the inclusion of multiple attestation variants, where at least one variant must be satisfied by an enclave in order to gain access. This provides an additional layer of security and verification. In the example provided, the MRENCLAVE value is used to identify the enclave measurement for attestation [10]. If the MRENCLAVE differs from the expected value, the attestation process will be aborted, and the application will be terminated without receiving any secrets or configurations from CAS. Alongside the attestation requirements, the service configuration includes the command to be executed, any environment variables needed, and the working directory from which relative paths are resolved.

The "secrets" section (lines 13-16) allows for the definition of secrets within the session. Secrets are uniquely identified by a name and can be generated by CAS or explicitly provided. Using CAS-generated secrets offers a great advantage in terms of security since their values are never disclosed to humans. These secrets can be easily integrated into services by injecting them into program arguments, environment variables, and files. To make this injection easier, a placeholder variable in the form of *$$SCONE::secret_name$$* can be used to reference the actual secrets, as seen in line 11. This placeholder is replaced with the actual secret value when the service starts, guaranteeing that only authorized entities can securely access and utilize sensitive information.

Sessions can be instantiated and managed using the SCONE CLI, a powerful tool for interacting with the SCONE platform. In addition to the *attest* command, which builds trust and attests a remote CAS, the *scone session* commands are available for session management. These commands allow you to validate written policies, update existing sessions, and verify if the active session aligns with a specific policy. The CLI provides a comprehensive set of tools for creating and maintaining sessions in SCONE.

In addition to attestation configurations and secrets management, SCONE offers a cross-compilation feature for Go applications to securely run in SGX enclaves without any changes to the source code. The Go cross-compiler from SCONE preserves the integrity and confidentiality of the code and data while maintaining the original Go source code. Converting the source code into an enclave-compatible format enables developers to deploy their applications without compromising security.

5.3 Execution Flow

The execution flow of the compliance certification system encompasses several key steps, ensuring a robust and secure process for evaluating and certifying Kubernetes clusters. From the compilation and attestation of the CLI and Marvin to the emission of compliance certificates on the Ethereum blockchain, each step is carefully designed to provide transparency, integrity, and trust in the certification process. Let's explore the detailed flow of the compliance certification system, highlighting the interactions and functionalities of each component involved.

To start the compliance certification process for a Kubernetes cluster, the user needs to request a scan. This will trigger the attestation process to verify the integrity of the CLI and Scanning Engine, assuming both have undergone prior audits. The attested CLI then obtains secrets from CAS and conveys relevant parameters and configurations to Marvin, indicating the target Kubernetes cluster to be scanned. Marvin operates within the TEE and conducts a comprehensive assessment of the compliance and security posture of the Kubernetes cluster. It generates test results within the TEE, ensuring that the scanning process remains confidential and secure. The attested CLI, also operating within the TEE, receives the test results generated by Marvin and analyzes them to identify any issues or vulnerabilities in the Kubernetes cluster and determine its compliance status. All analysis and decision-making procedures occur within the TEE. If all tests are successful, the attested CLI triggers the execution of a smart contract on the Ethereum blockchain using the credentials received from CAS. Finally, the certificate is issued, providing a TxID and data certificate-related data.

6. IMPLEMENTATION STATUS

Before delving into the implementation status of the compliance certification system, it is essential to consider the specific environment and assumptions in which the implementation has taken place. A private cloud virtual machine with Intel SGX support has been utilized in the current setup. This environment offers a controlled and secure testing ground for the system but comes with inherent scalability and real-world deployment limitations. It is important to acknowledge the challenges involved in integrating technologies such as Trusted Execution Environments and Blockchain. These technologies require meticulous consideration of their complexities and intricacies

The compliance certification system implementation is underway. Several significant milestones have been achieved thus far. The CLI has been successfully integrated with the scanning engine and blockchain functionality. Furthermore, the system's end-to-end execution flow, which encompasses critical processes such as attestation and secure execution within the TEE, has been executed successfully. These achievements demonstrate the substantial progress towards realizing a robust and secure compliance certification solution. However, it is worth noting that the automation resources necessary to replicate the system at scale have yet to be fully developed. Additionally, while the smart contracts have been tested using Ganache, a local blockchain simulator, they have not been deployed in a real-world blockchain network.

7. EVALUATION AND RESULTS

During the evaluation of the proof of concept, which followed the designed model described in section 4, it demonstrated the ability to address the presented challenges and sufficiently resolve the concerns outlined in section 3.3. These concerns included the trustworthiness of the scanning process, potential adulterated compliance checks, lack of transparency, the need for trust in the scanning tool, and mitigating reliance on trust.

However, it is crucial to consider certain factors when building a production-ready solution. Establishing a reliable association between the issued certificate and the verified cluster should be an important consideration to ensure the certification process's authenticity and trustworthiness. Without an association, the cluster owner may be able to manipulate the scanning process by maintaining various clusters with different security postures and forwarding the scan target to a healthy cluster. A possible solution to help address this problem is to use stable and public API addresses if the scanning tool is run remotely, preventing manipulation across different clusters.

While there are certain aspects to consider when transitioning to a robust and production-ready solution, the findings from the proof of concept evaluation highlight its effectiveness in addressing the identified concerns. It successfully demonstrated its capability to tackle the challenges outlined in the previous sections. Nevertheless, it is important to note the key considerations for a production-ready solution. By addressing these aspects and implementing necessary enhancements, the system can be further implemented into a reliable and scalable solution for Kubernetes cluster security.

8. FUTURE WORK

As the compliance certification system for Kubernetes clusters continues to evolve, several areas of future work can be explored to enhance its capabilities and address emerging security challenges. These areas include the following:

I. Improved Certificate Association: Ensuring a reliable and secure association between issued certificates and verified infrastructure is essential. To achieve this, future efforts should be made to develop mechanisms that strengthen the association between certificates and specific clusters. This will improve the reliability and credibility of the certification process.

II. Integration with Real-World Blockchain Networks: While the proof of concept utilized a local blockchain simulator for testing, integrating the compliance certification system with real-world blockchain networks, such as Ethereum, can provide added security and immutability benefits. This would involve deploying and interacting with smart contracts on public or private blockchain networks.

III. Scalability and Automation: To handle large-scale Kubernetes clusters effectively, it's important to develop automation resources that enable the system to scale. This includes optimizing resource utilization, parallelizing scanning processes, and integrating with orchestration systems.

Furthermore, regularly updating the system with the latest security guidelines and best practices will ensure its effectiveness in identifying and mitigating new risks. Prioritizing these areas for future development will enable the compliance certification system to become a strong and thorough solution for safeguarding Kubernetes clusters, ultimately giving organizations greater assurance regarding their infrastructure's security.

9. ACKNOWLEDGMENTS

First and foremost, I would like to express my gratitude to God for His guidance and blessings in my life. I would also like to thank my family and partner for their unwavering support throughout this journey. To my friends, thank you for your camaraderie and partnership during the entire process.

This work has been supported by the Distributed Systems Laboratory (LSD) at the Federal University of Campina Grande, which provided the infrastructure for testing and implementation.

Lastly, I would like to extend my appreciation to my advisor, Andrey Brito, for his guidance and support throughout this work. His contribution to the successful completion of my thesis has been invaluable.

10. REFERENCES

[1] CNCF Annual Survey 2022. Cloud Native Computing Foundation. Retrieved from https://www.cncf.io/reports/cncf-annual-survey-2022.

[2] CNCF Annual Survey 2021. Cloud Native Computing Foundation. Retrieved from https://www.cncf.io/reports/cncf-annual-survey-2021.

[3] VMware. 2023. State of Kubernetes 2023. Palo Alto, CA: VMware. (Ebook). Retrieved from https://tanzu.vmware.com/content/ebooks/stateofkubernetes-2023.

[4] M.S. Shamim, F.A. Bhuiyan, and A.A. Rahman. 2020. XI Commandments of Kubernetes Security: A Systematization of Knowledge Related to Kubernetes Security Practices. In Proceedings of the 2020 IEEE Secure Development (SecDev), 58-64.

[5] Strom, B. E., Applebaum, A., Miller, D. P., Nickels, K. C., Pennington, A. G., & Thomas, C. B. (2018). Mitre att&ck: Design and philosophy. In Technical report. The MITRE Corporation.

[6] National Security Agency, Cybersecurity and Infrastructure Security Agency. (2022). Kubernetes hardening guidance v 1.2. Retrieved from https://media.defense.gov/2022/Aug/29/2003066362/-1/-1/0/CTR_KUBERNETES_HARDENING_GUIDANCE_1.2_20220829.PDF

[7] M. Sabt, M. Achemlal, and A. Bouabdallah. 2015. Trusted Execution Environment: What It is, and What It is Not. In Proceedings of the 2015 IEEE Trustcom/BigDataSE/ISPA (TrustCom), Helsinki, Finland, 57-64. DOI: 10.1109/Trustcom.2015.357.

[8] Nofer, M., Gomber, P., Hinz, O., et al. 2017. Blockchain. Business & Information Systems Engineering 59, 3, 183-187. DOI: 10.1007/s12599-017-0467-3.

[9] Szabo, N. 1997. Formalizing and Securing Relationships on Public Networks. First Monday 2, 9. DOI: 10.5210/fm.v2i9.548.

[10] Costan, V., & Devadas, S. 2016. Intel SGX explained. Cryptology ePrint Archive.

[11] Arnautov, S., Trach, B., Gregor, F., Knauth, T., Martin, A., Priebe, C., Lind, J., Muthukumaran, D., O'keeffe, D., Stillwell, M., & others. 2016. Scone: Secure Linux Containers with Intel SGX. In OSDI, 16, 689-703.

[12] Dannen, C. 2017. Introducing Ethereum and Solidity. DOI: 10.1007/978-1-4842-2535-6.