Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.056141

ARTICLE

NCCMF: Non-Collaborative Continuous Monitoring Framework for Container-Based Cloud Runtime Status

Tao Zheng[1], Wenyi Tang[1,2,4,*], Xingshu Chen[1,3,4] and Changxiang Shen[1,3,4]

[1]School of Cyber Science and Engineering, Sichuan University, Chengdu, 610065, China

[2]Intelligent Policing and National Security Risk Management Laboratory, Luzhou, 646000, China

[3]Cyber Science Research Institute, Sichuan University, Chengdu, 610065, China

[4]Key Laboratory of Data Protection and Intelligent Management (Sichuan University), Ministry of Education, Chengdu, 610065, China

*Corresponding Author: Wenyi Tang. Email: wtang@scu.edu.cn

Received: 15 July 2024 Accepted: 18 September 2024 Published: 15 October 2024

ABSTRACT

The security performance of cloud services is a key factor influencing users' selection of Cloud Service Providers (CSPs). Continuous monitoring of the security status of cloud services is critical. However, existing research lacks a practical framework for such ongoing monitoring. To address this gap, this paper proposes the first Non-Collaborative Container-Based Cloud Service Operation State Continuous Monitoring Framework (NCCMF), based on relevant standards. NCCMF operates without the CSP's collaboration by: 1) establishing a scalable supervisory index system through the identification of security responsibilities for each role, and 2) designing a Continuous Metrics Supervision Protocol (CMA) to automate the negotiation of supervisory metrics. The framework also outlines the supervision process for cloud services across different deployment models. Experimental results demonstrate that NCCMF effectively monitors the operational state of two real-world IoT (Internet of Things) cloud services, with an average supervision error of less than 15%.

KEYWORDS

Container-based cloud; non-collaborative; continuous monitor; runtime status

1 Introduction

The security of cloud services is of paramount importance in light of the increasing prevalence of cloud-based operations and data storage, as well as the rising user expectations for robust security and reliability [1]. However, traditional security measures are ineffective in the complex and dynamic environment of container-based cloud services. Static security assessments and one-time security policies are insufficient to address the evolving security needs of container cloud environments [2]. In contrast, continuous monitoring provides continuous and comprehensive security monitoring to ensure the long-term security of container cloud services. Continuous monitoring of the operational status of container cloud services involves regular and continuous monitoring and evaluation of the operational status of container cloud services within a cloud computing environment [3]. The

Copyright © 2024 The Authors. Published by Tech Science Press.
This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

implementation of continuous monitoring enables users to obtain pertinent information regarding the security status of their container cloud services in a timely manner. This encompasses a range of aspects, including access control, data isolation, and vulnerability scanning. Such monitoring enables the identification of potential security risks and facilitates the implementation of appropriate countermeasures, thereby protecting business operations and sensitive data. Furthermore, continuous monitoring enables users to fulfill their compliance and regulatory obligations. By continuously monitoring container cloud services, users can ensure compliance with relevant regulations and standards and mitigate potential legal risks [4].

Currently, numerous research efforts have been undertaken globally to address the challenges associated with security assessment and regulation of container-based cloud services. These studies primarily focus on two key areas: cloud service security assessment modeling and the development of operational regulatory standards. In the field of security assessment, researchers endeavor to utilize existing cloud service audit logs, identify assessment indicators based on specified criteria in relevant standards, and subsequently produce numerical assessment results through suitable statistical methods. Regarding standard development, international organizations and standardization bodies are actively engaged in formulating standards and specifications applicable to the continuous monitoring of cloud service operational status. These standards aim to provide a unified framework for ensuring ongoing evaluation and monitoring of the security and performance of cloud services. However, current standards development primarily emphasizes the creation of standards themselves, with limited consideration given to the specific implementation and operationalization frameworks. Consequently, a gap exists in terms of an operational monitoring framework [5] for continuous monitoring of cloud service operational status [6]. In practical applications, users often encounter challenges such as selecting appropriate monitoring tools and techniques, establishing a monitoring indicator system, and conducting real-time monitoring and troubleshooting. Furthermore, the complexity and diversity of cloud services amplify the monitoring difficulties, necessitating the design of tailored monitoring strategies for different deployment models and service types. Additionally, reducing subjectivity in the monitoring process is an important concern [7]. Therefore, the key challenges that must be addressed in the framework for continuous monitoring of cloud service operational status are as follows:

Challenge 1: Clarifying the delineation of responsibilities between regulators and cloud service providers during the monitoring process for different types of cloud services to define the selection scope of monitoring indicators.

Challenge 2: Designing a practical continuous monitoring agreement that enables automated updates of monitoring indicators and results to enhance the operational efficiency of the monitoring framework.

Challenge 3: Develop a dynamically updatable feedback mechanism for monitoring results to assist cloud service providers in real-time corrective actions.

To address the above challenges, we design and implement ***NCCMF***, a non-collaborative container-based cloud operation state continuous monitoring framework. Specifically, ***NCCMF*** firstly clarifies the operation monitoring roles and responsibilities of cloud service providers (CSP) and continuous supervisors (CMO) based on the "*GB/T 31167-2023*" standard. Second, it clarifies the content and scope of continuous monitoring indicators based on "*GB/T 31168-2023*", and realizes the fusion of multi-source indicators through D-S theory so as to enhance its objectivity. Next, it designs an automated continuous monitoring protocol CMA based on the cloud service level indicators specified in the "*GB/T 36325-2018*" standard. Finally, ***NCCMF*** complements and extends the framework specified in the "*GB/T 37972-2019*" standard. In this paper, we prove the security of the continuous

monitoring agreement (CMA) through formal validation methods, and also carry out experiments in two real IoT cloud services, and the results show that the ***NCCMF*** framework is able to effectively complete the continuous monitoring of container cloud service operation status.

In summary, our contributions are as follows:

1. A continuous monitoring framework for non-collaborative container cloud operation status is proposed, which is based on cloud service security standards and carries out continuous monitoring from three dimensions: security capability, security assessment and service level.
2. To avoid monitoring content redundancy, a scalable indicator system for continuous monitoring of container-based cloud service operation status is established, and the potential dependency between multi-source indicators is eliminated by using hierarchical analysis, while the fusion of indicator matrices is realized based on the D-S evidence theory to avoid indicator dependency conflicts and the influence of human subjective factors.
3. Experimental analyses via two real IoT cloud services were carried out to prove the security of the effectiveness of the ***NCCMF*** framework.

The rest of this paper is as follows: Section 2 reviews related work. Section 3 introduces our designed and implementation of ***NCCMF***. Section 4 discusses performance evaluation. Finally, Section 5 concludes our work.

2 Related Work

2.1 Cloud Service Monitoring

The research on continuous monitoring of cloud service operational status is currently in an active phase of development, with numerous related studies emerging both domestically and internationally. To address the challenges of incomplete coverage and redundant monitoring in cloud services, Srinivas et al. [3] introduced a data-driven intelligent monitoring framework, which recommends appropriate monitoring strategies based on cloud service attributes. Building upon this, a deep learning-based framework has been proposed to further optimize monitoring recommendations by leveraging service attributes. Akhbarifar et al. [4] developed a secure remote health monitoring model aimed at disease diagnosis within a cloud-based IoT environment. Similarly, Sundas et al. [5] proposed SPMR, a smart patient monitoring and recommendation system utilizing cloud analytics and deep learning. Bonci et al. [6] explored the use of Ultra Wide Band (UWB) communication for condition-based monitoring, discussing its potential applications in Industry 4.0 and demonstrating the implementation of a UWB sensor network for machine vibration monitoring. Ruiz-Zafra et al. [7] introduced NeoCam, an edge-cloud platform designed for non-invasive real-time monitoring in neonatal intensive care units. Additionally, Soveizi et al. [8] reviewed the security and privacy challenges associated with cloud-based scientific and business workflows. Further contributions include cloud system monitoring methods employing deep learning and system logs, as proposed by Xu et al. [9] and Rajadurai [10].

2.2 Cloud Service Assessment

Cloud service assessment involves a comprehensive evaluation of cloud services to determine their attributes and capabilities in areas such as security, reliability, performance, and compliance. Wang et al. [11] proposed an accurate cloud service quality assessment method and Qu et al. [12] designed a cloud service assessment scheme based on subjective and objective assessment. Xiao et al. [13] further improved the assessment model for evaluating candidate design schemes by

utilizing an interval rough integrated cloud model in an uncertain group environment. Sen et al. [14] developed an offline risk assessment for cloud service providers (CSP), while Liu et al. [15] introduced a quality assessment method for point clouds through multi-view projection. Liu et al. [16] and Wang et al. [17] also proposed quality assessment approaches for point clouds. Additionally, Zhang et al. [18] and Sen et al. [19] investigated risk assessment models for cloud Petri nets and application design. Parast et al. [20] conducted a review of cloud computing security within service-based models.

3 The Proposed NCCMF

3.1 Overview

The meanings of the symbols used in this paper are shown in Table 1. As shown in Fig. 1, ***NCCMF*** contains two participants, CSP and CMO, which is based on "*GB/T 37972-2019*" operational regulatory framework, the "*GB/T 31167-2023*" as the basis for the construction of CSP supervisory metrics, and the "*GB/T 31168-2023*" as the benchmark for CMO to assess the security capability of the CSP, and at the same time, draws on "*GB/T 36325-2018*" for the negotiation of supervisory level and elimination of metrics' subjectivity, and ultimately establishes a fine-grained non-collaborative cloud service operational status continuous monitoring framework. The ***NCCMF*** has three modules, the consultation module, which defines the responsibilities and roles of the participants, constructs the monitoring indicator system and negotiates the scope of the monitoring. The monitoring module accomplishes continuous operation monitoring through CMA. The feedback module realizes the feedback of the monitoring results and updates the consultation at the same time.

Table 1: Meaning of the symbols

Symbols	Notions
A	Indicator judgment matrix
a	Comparative importance
w	Weight vector
θ_{max}	Maximum eigenvalue of the matrix
CR	Consistency ratio
CI	Consistency index
RI	Average stochastic consistency index
m	Confidence function

The process of continuous monitoring by the CMO over CSP is illustrated in Fig. 2. The primary objective of this supervision is to ensure that CSP consistently implement the security measures they have committed to regarding cloud services, fulfill the security supervision responsibilities and obligations stipulated in the "*GB/T 31167-2023*", and adhere to the security capability requirements outlined in the "*GB/T 31168-2023*". The verification process involves the CMO examining the self-certification materials provided by the CSP in a non-collaborative manner. CSP are required to establish and maintain a local monitoring platform or component to oversee the cloud computing platform. They must ensure the security of their cloud platform by developing and implementing regulatory strategies, conducting periodic risk assessments, and monitoring their platform. This

includes continuously collecting and analyzing metric information and providing ongoing internal information security education.

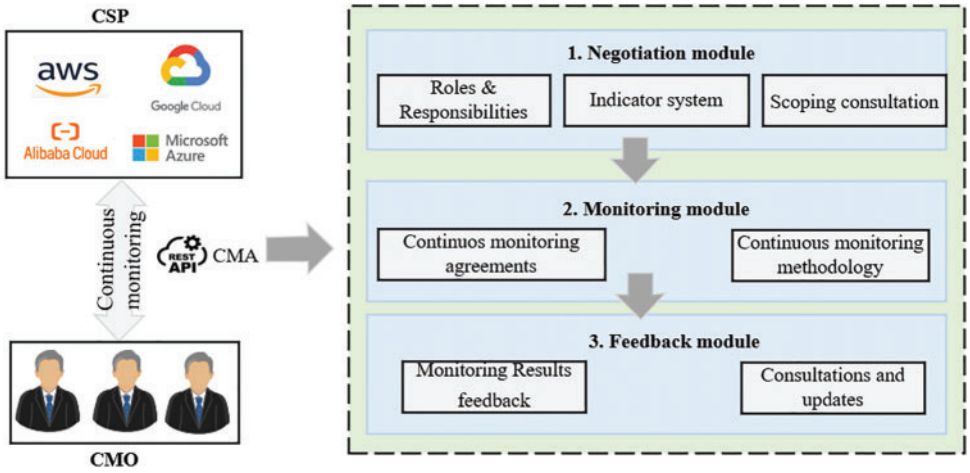

Figure 1: The workflow of NCCMF

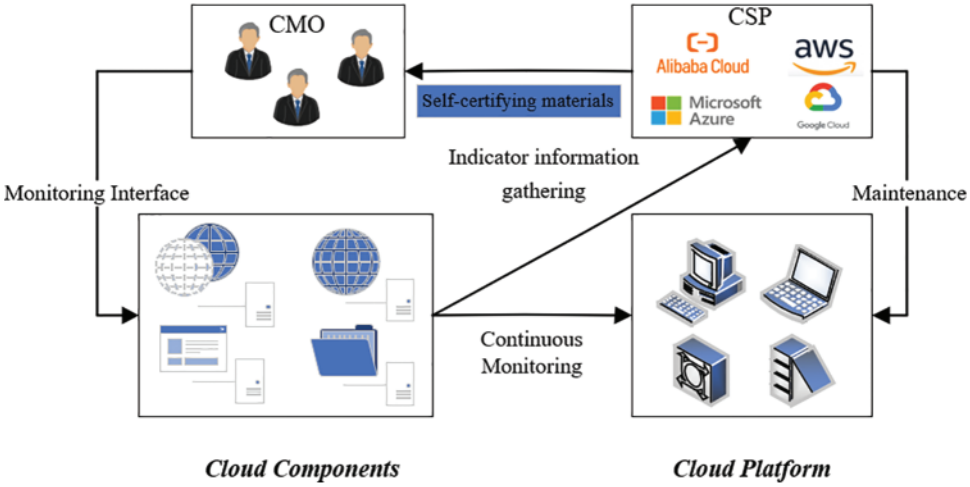

Figure 2: Schematic diagram of the non-collaborative continuous monitoring process

To achieve self-certification, CSPs submit reports to the CMO detailing their regulatory strategies, the outcomes of these strategies, and the methods and results of risk assessments and testing. The self-certification materials must be organized according to the format and indicators specified by the CMO and submitted at a frequency agreed upon with the CMO. Additionally, CSP must provide a monitoring interface to the CMO, enabling access to relevant data and information of the cloud platform for the purpose of validating the accuracy of the self-certification materials. If the discrepancy between the CMO's verification results and the CSP's self-certification materials falls within an acceptable range (as negotiated between the CMO and the CSP), it confirms that the CSP meets the CMO's regulatory requirements. Otherwise, the CSP must provide timely, dynamic feedback and rectify any issues in accordance with the requirements set forth in the "*GB/T 31168-2023*" until compliance is achieved.

3.2 Standards-Oriented Monitoring Consultations

To address **Challenge 1**, the ***NCCMF*** employs "*GB/T 31168-2023*" as the guiding standard for the security capability dimension, the "*GB/T 31167-2023*" for the security indicator dimension, and "*GB/T 36325-2018*" for the security level dimension. As illustrated in Fig. 3, the ***NCCMF*** defines the supervision scope and indicator granularity based on these three standards. It delineates the security responsibility model for different cloud service deployment modes and specifies the corresponding supervision levels.

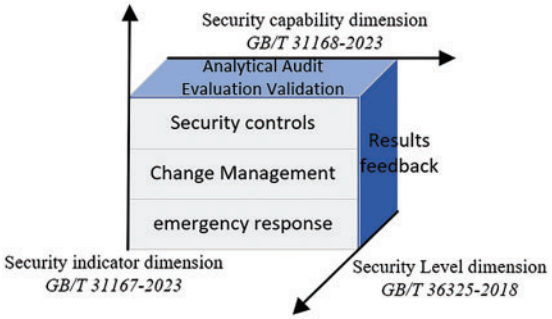

Figure 3: Prototype diagram of the 3D model for continuous runtime monitoring

Second, to prevent redundant supervision, the ***NCCMF*** develops the cloud service operation status supervision indicator system as depicted in Table 1, based on the "*GB/T 37972-2019*". To ensure the objectivity of indicator attributes, the ***NCCMF*** employs a fusion method that integrates D-S evidence theory and hierarchical analysis. This approach minimizes indicator dependency conflicts and reduces the influence of human subjectivity in calculating indicator weights, thereby strengthening the "non-collaborative" nature of the supervision process. An indicator judgment matrix is created for the hierarchy as follows:

$$\mathrm{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \cdots & \cdots & \cdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \tag{1}$$

In this context, a_{mn} represents the comparative importance between indicator m and indicator n. A nine-point scale is utilized for quantitative assessment: $a_{mn} = 1$ indicates that both indicators are equally important. $a_{mn} = 3$ indicates that m is slightly more important than n. $a_{mn} = 5$ indicates that m is significantly more important than n. $a_{mn} = 7$ indicates that m is much more important than n; and $a_{mn} = 9$ indicates that m is extremely more important than n. Values of $a_{mn} = 2, 4, 6, 8$ signify intermediate levels of importance between these judgments.

Second, the vector of weights between the indicators is calculated as:

$$|\omega_i| = \sqrt[n]{\Delta}, \Delta = \prod_{j=1}^{n} a_{mn} \tag{2}$$

Next, the weight vector is normalized as $\omega_i = \dfrac{|\omega_i|}{\sum_{i=1}^{n} |\omega_i|}$ and the maximum eigenvalue of the matrix is calculated as $\theta_{max} = \dfrac{1}{n} \sum_{i=1}^{n} \dfrac{(A\omega)_i}{\omega_i}$, where A is the judgment matrix. It can be observed that if matrix A satisfies $a_{mn} = a_{m\omega}/a_{n\omega}$, then there exists a unique non-zero maximum eigenvalue for matrix A. This condition implies that the matrix exhibits complete consistency, meaning that indicator dependence

disappears, and the indicators become entirely independent. However, due to the inevitable influence of human factors in the process of continuous monitoring, it is impossible to completely eliminate the dependence between indicators. Therefore, a consistency test of the matrix is necessary to minimize the excessive error introduced by human factors. The consistency ratio $CR = CI/RI$ can estimate the overall consistency of the judgment matrix. Here, the consistency index CI is defined as $CI = \frac{\theta_{max} - n}{n-1}$, where a larger CI indicates poorer consistency of matrix A, necessitating adjustments to the elements of the judgment matrix. The random index RI is the average stochastic consistency index, which is randomly generated by matrices of size n. It is generally accepted that the judgment matrix A has good consistency when $CR < 0.1$, otherwise, the matrix should be adjusted.

Indicator fusion based on D-S evidence theory. In D-S evidence theory, a proposition T is considered an element of a recognition framework φ. If $\forall \mathrm{m}(T) > 0$, T is regarded as a focal element of a confidence function F. Given two confidence functions F_1 and F_2 on the same recognition framework φ, with m_1 and m_2 as the corresponding basic confidence assignments, the process works as follows:

$$m(\varphi) = \sum\nolimits_{T_i \cap H_j = T} m_1(T_i)\, m_2\left(H_j\right) \tag{3}$$

To further mitigate the impact of indicator dependency conflicts on the results, the indicator weights ω_i obtained from hierarchical analysis are incorporated into the evidence source. Assuming the weight vector satisfies $\omega_i \in [0,1]$ and $\sum_{i=1}^{n} \omega_i = 1$, the relative weights $W = (\omega_1 \omega_2 \ldots \omega_n)/\omega_{max}$ are derived from the equation $1 - \delta_i = \omega_i/\omega_{max}$. Thus, the confidence function of evidence theory is modified to:

$$m^k(T) = (1 - \delta_i)\, m(T) \tag{4}$$

where $m(T)$ is the original confidence function. This modification adjusts the confidence assigned to each proposition T by accounting for the relative weights of the indicators, thus reducing the influence of indicator dependency conflicts. Therefore, the total confidence assigned to T can be expressed as:

$$m^k(\varphi) = \sum\nolimits_{\cap T_i = T} \prod\nolimits_{i=1}^{n} \left[(1 - \delta_i)\, m_i(T_i) + \delta_i\right] \tag{5}$$

At this stage, the new evidence theoretical synthesis formula incorporating the weight of each indicator is obtained. This formula allows the CMO and CSP to further refine the scope of the monitoring indicators and the corresponding content based on the calculation results. This process aims to minimize human interference factors as much as possible, ensuring a more objective and reliable supervision framework.

3.3 Continuous Monitoring with CMA

To address **Challenge 2**, as is shown in Table 2, ***NCCMF*** establishes a system of indicators for continuous monitoring of the operational status of cloud services based on the guidance of the three standards.

Table 2: Indicators for continuous monitoring of cloud services

Security capacity-A	Type-B	Monitor indicator-C
Security control indicator-A1	*Identity and authorization management-B1*	*Authentication strength-C1*

(Continued)

Table 2 (continued)

Security capacity-A	Type-B	Monitor indicator-C
		Authorization to design-C2 *Effectiveness and compliance-C3*
	Separation of privileges and least privilege-B2	*Role Assignment and Privilege-C4* *Principle of Least Authority-C5* *Permission Change Recovery-C6*
Change management indicator-A2	*Data encryption and backup changes-B3*	*Data classification-C7* *Data encryption-C8* *Data backup-C9*
	Cloud Platform Release Change-B4	*Cloud Platform Version-C10* *Cloud service running body-C11* *Cloud Platform Architecture-C12*
Emergency response indicator-A3	*Security event and threat monitoring-B5*	*Real-time monitoring reporting-C13* *Threat intelligence analysis-C14* *Security incident disposal-C15*
	Log management audit-B6	*Log collection and storage-C16* *Log analysis-C17* *Audit and compliance-C18*

To facilitate the continuous supervision of the cloud service's operational state by the CSP, the ***NCCMF*** implements a negotiable and configurable CMA for overseeing the cloud service's security state, as illustrated in Fig. 4.

As shown in Algorithm 1, based on the RESTful API protocol architecture, CMA should be universal, simple, easy to implement, and interactive to develop interface specifications. CSPs provide interfaces according to these specifications. Interfaces can be grouped into collections based on their functions, which can serve as functional components. The CMA protocol can achieve: 1) Describing supervisory data indicators, interface specifications, data collection methods, frequency, and format between CMO and CSP. 2) Defining transmission specifications, including active/passive methods and data encryption. 3) Automating negotiation of supervisory data content and scope between CMO and CSP. 4) Enabling continuous supervision of the cloud platform by CMO.

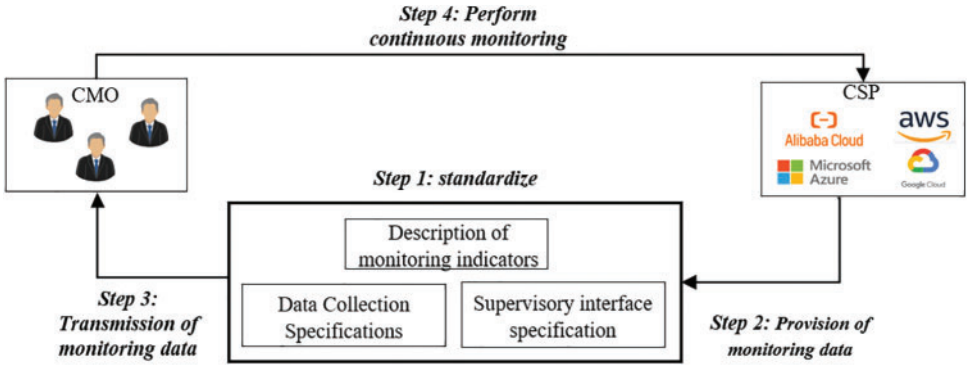

Figure 4: The proposed CMA

Algorithm 1: Continuous monitoring agreement

Input: *Regulatory data*
Output: *Operational status feedback*

Step 1: standardize phase
Supervise the description of data-related metrics
(security, availability, performance, and other metrics)
Define the interface and data format between CMO and CSP
Define the collection mode, frequency and specification of supervisory data
Determine how surveillance data will be transmitted and encrypted
Step 2: Provision of monitoring data
CSP provides monitoring data as specified in ***Step 1***
(Login audit logs, access control logs, security event logs)
Step 3: Transmission of monitoring data
The CSP transmits the collected supervisory data to the CMO using the transmission and encryption methods specified in the CMA protocol. Encryption algorithms and protocols are used to ensure the confidentiality and integrity of the data during the data transmission process.
Step 4: Perform continuous monitoring
CMO ensures ongoing oversight of the cloud platform through the supervisory interface, maintaining the security of CSP throughout their operational cycle. The CMO regularly communicates and consults with CSP, updating the content and scope of supervisory data to meet evolving security needs.
Return the monitoring results

Among them, the description of supervisory indicators is based on the analysis of supervisory needs, specifying the content and scope of supervision for the supervisory object. The supervision interface specification regulates the data acquisition interface provided by the supervisory object to the supervisory agency, ensuring efficient and accurate data collection. It also ensures that the interface is user-friendly and scalable to facilitate supervision content negotiation. The data acquisition specification standardizes the operation of cloud service data acquisition by the supervisory body, ensuring process compliance, scope accuracy, and scalable acquisition frequency.

3.4 Dynamically Updated Results Feedback

Monitoring Feedback. Regulators provide continuous monitoring and feedback (e.g., satisfaction scores) on the security of the operational state of the cloud service based on the quality of their experience (i.e., subjective perceptions regarding the security attributes of the cloud service) during the use of the real cloud service.

Self-Monitoring Information. The cloud provider selfmonitors the service capabilities (security capabilities and quality of service) of its service offerings against the metrics defined by the Continuous Monitoring Framework and provides the values of the metrics (i.e., committed service objectives) that it is able to maximize the satisfaction of and quantify.

Evidence Deliverables. According to the description of the framework, the cloud service provider submits deliverables that can prove that the operational status of its cloud services continuously meets the security requirements, including but not limited to various documents, pictures, audio-visuals, and data and other supporting materials, in accordance with the monitoring requirements of the continuous monitoring framework.

Dynamic Feedback. The CMO and CSP regularly analyze and consult based on the results of ongoing supervision and execute the previously mentioned process. This dynamic feedback process aims to continuously update the scope and content of supervision, thereby enhancing the continuous oversight of the cloud service's operational status.

4 Experiment

To verify the effectiveness of the proposed continuous monitoring framework, this paper chooses the "cloud-edge-device" fusion scenario for continuous monitoring of cloud service security status. Specifically, the CloudIoT system collects data from sensor end devices, uploads the data to the cloud through the sensor network for storage and analysis, and returns the cloud calculation results to the device side.

4.1 Establishing Monitoring Indicators

As depicted in Table 3, in tandem with the comprehensive metrics system for continuous supervision outlined in Section 3.3, the metrics have been adapted with a focus on the ongoing monitoring of the security state. This adaptation takes into account the data flow diagram of the SDN architecture and the assessment metrics system centered around data security. Notably, this system exhibits characteristics of both IaaS services and PaaS and SaaS services. Additionally, the integration of IoT-generated data from diverse sources enhances the framework's ability to meet the continuous supervision requirements.

Table 3: Indicators for continuous monitoring of CloudIoT services

Security capacity-A	Type-B	Monitor indicator-C
Device-aware layer-A1	*Access control-B1*	*Authentication-C1* *Access control-C2* *Firmware integrity-C3*

(Continued)

Table 3 (continued)

Security capacity-A	Type-B	Monitor indicator-C
	Device status-B2	*Firmware patch-C4* *Hardware security-C5* *IDS and firewalls-C6*
Platform network layer-A2	*Sensor network security-B3*	*Socket security-C7* *Interface security-C8* *Port security-C9*
	Network traffic security-B4	*Network traffic-C10* *Device logging-C11* *Device traffic-C12*
Application layer-A3	*Virtual appliance security-B5*	*Virtual storage image security-C13* *Virtual border capabilities-C14*
	Virtual network security-B6	*DNS service device security-C15* *Virtual switch security C-16* *Malicious cyber attack C-17*
	Data Security-B7	*Data integrity and segregation-C18* *Data confidentiality measures-C19* *Data destruction management-C20*

In the device sensing layer A1, communication data is realized with the help of wireless transmission technology such as Bluetooth for end device data transmission. While IoT protocols tend to consider performance overheads first and foremost, therefore authentication and access control policies B1 and device security status B2 are necessary to be considered. Around these two types of issues, this experiment collects six types of evidence metrics: authentication mechanism C1, access control policy C2, firmware integrity verification C3, firmware patch level C4, device hardware security measures C5 and IDS and firewall C6.

At the platform network layer A2, the control and data layers of the SDN-based sensing network are separated, and the data layer forwards packets according to the control layer at the controller using the OpenFlow protocol. Therefore, in order to protect the data security of layer A2, the sensing network security B3 and network traffic security B4 are necessary to be considered. Further analysis shows that the following six metrics need to be considered: socket security C7, network interface security C8, network port security C9, network traffic between edge devices and cloud services C10, cloud environment network device logs C11, and sensing network device traffic C12.

At the application layer A3, since the most important security factor for the cloud in cloud-side-end IoT scenarios is virtual security, improper virtualization isolation strategies can have a serious

impact on data security. Therefore, virtual device security B5, virtual network security B6 and program data security B7 are the factors to be considered. Further considering the data security factors, this chapter collects the following eight evidence indicators: virtual storage image security C13, virtual boundary capability C14, DNS service device security C15, virtual switch security C16, malicious network attack C17, data integrity and isolation C18, data confidentiality measures C19 and data destruction management C20.

4.2 Validity Analysis of NCCMF

4.2.1 Experimental Scenarios

To further validate the effectiveness of the continuous supervision framework, this chapter demonstrates the continuous supervision process of the supervision framework using two real IoT cloud service systems, Google Brillo and Azure IoT Suite. First, we introduce the two systems: Google Brillo is an Internet of Things (IoT) operating system launched by Google. It aims to provide a lightweight, secure, and reliable operating system for connecting and managing IoT devices. Brillo's goal is to simplify the process of developing and deploying IoT devices and provide developers with better security and interoperability. Brillo is based on the core of the Android operating system, but with unnecessary components and features removed, which allows Brillo to run on resource-constrained devices. Azure IoT Suite is a comprehensive set of Internet of Things (IoT) solutions from Microsoft. It is designed to help organizations easily build, deploy and manage IoT solutions for device connectivity, data collection and analysis, real-time monitoring and remote control. It provides key features such as connectivity, data collection and analysis, real-time monitoring and remote control. By using Azure IoT Suite, organizations can easily build secure and reliable IoT solutions and leverage the power of the cloud platform for smarter and more efficient business applications.

4.2.2 Experimental Results

Evidence corresponding to each of the evidence indicators proposed in this paper can be found in the official documents of the two services, thus demonstrating the ways in which these service providers cope with the security issues, as well as the specific measures and enhancements taken. Based on the list of indicators constructed in the previous section, the CMO analyzes the weights of the evidence indicators based on the extracted textual evidence and combines the hierarchical indicator fusion method in Section 3.2 to evaluate the continuous monitoring of the security status of the two systems.

The results of the experiment are presented in Table 4, which demonstrates that the assessment results of Azure IoT Suite (21.1869) are marginally higher than those of Google Brillo (21.0634). Furthermore, the discrepancy between the results and the average value (21.4621) is within an acceptable range, thereby substantiating the practicality and efficacy of the framework.

Table 4: Results of security state monitoring of two real CloudIoT systems

Indicator number	Evidence list	Ranking of monitoring results		
		Google Brillo	Azure IoT Suite	Average
A1-B1-C1	*Authentication*	12	17	21
A1-B1-C2	*Access control*	15	16	23
A1-B1-C3	*Integrity*	13	20	18

(Continued)

Table 4 (continued)

Indicator number	Evidence list	Ranking of monitoring results		
		Google Brillo	Azure IoT Suite	Average
A1-B2-C4	*Firmware patch*	15	18	16
A1-B2-C5	*Hardware*	17	19	17
A1-B2-C6	*IDS firewalls*	16	10	16
A2-B3-C7	*Socket security*	5	2	13
A2-B3-C8	*Interface*	7	3	15
A2-B3-C9	*Port*	9	7	21
A2-B4-C10	*Edge traffic*	12	10	9
A2-B4-C11	*Device logging*	10	8	12
A2-B4-C12	*Sensing traffic*	6	4	3
A3-B5-C13	*Virtual images*	13	9	7
A3-B5-C14	*Virtual border*	17	11	18
A3-B6-C15	*DNS device*	15	12	9
A3-B6-C16	*Interaction*	16	13	22
A3-B6-C17	*Malicious*	21	14	16
A3-B7-C18	*Data isolation*	19	13	14
A3-B7-C19	*Data measures*	13	5	5
A3-B7-C20	*Data destruction*	18	10	11
Score		21.0634	21.1869	21.4621

5 Conclusion

In this paper, we put forth a novel regulatory indicator fusion approach based on a hierarchical analysis method and D-S evidence theory. Furthermore, we have designed and implemented ***NCCMF***, a fine-grained continuous supervision framework for the operational status of cloud services. First, the ***NCCMF*** elucidates the security and regulatory responsibilities of each role in the cloud service environment. Second, the ***NCCMF*** enables users to comprehend the real-time status of cloud services through the continuous supervision process and assesses the security capabilities and security risks of the services provided by cloud service providers, thereby ensuring the continuous safeguarding of business and data security in the cloud environment. In future work, we will further explore how this method can continuously monitor the security state of target services in real scenarios under cloud-side-end architecture. In addition, we will also conduct research around performance with a view to reducing additional system overhead. We will consider introducing formal verification methods to carry out security analysis and proof of CMA protocols in our future research.

Acknowledgement: We thank the anonymous reviewers for their useful comments.

Funding Statement: This work was supported in part by the Intelligent Policing and National Security Risk Management Laboratory 2023 Opening Project (No. ZHKFYB2304), the Fundamental Research Funds for the Central Universities (Nos. SCU2023D008, 2023SCU12129), the Natural

Science Foundation of Sichuan Province (No. 2024NSFSC1449), the Science and Engineering Connotation Development Project of Sichuan University (No. 2020SCUNG129), and the Key Laboratory of Data Protection and Intelligent Management (Sichuan University), Ministry of Education.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Tao Zheng; data collection: Wenyi Tang; analysis and interpretation of results: Xingshu Chen; draft manuscript preparation: Changxiang Shen. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, Wenyi Tang, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] B. Alouffi, M. Hasnain, A. Alharbi, W. Alosaimi, H. Alyami and M. Ayaz, "A systematic literature review on cloud computing security: Threats and mitigation strategies," *IEEE Access*, vol. 9, pp. 57792–57807, 2021. doi: 10.1109/ACCESS.2021.3073203.

[2] R. R. Karn, P. Kudva, H. Huang, S. Suneja, and I. M. Elfadel, "Cryptomining detection in container clouds using system calls and explainable machine learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 3, pp. 674–691, 2021. doi: 10.1109/TPDS.2020.3029088.

[3] P. Srinivas, F. Husain, A. Parayil, A. Choure, C. Bansal and S. Rajmohan, "Intelligent monitoring framework for cloud services: A data-driven approach," in *Proc. 2024 IEEE/ACM 46th Int. Conf. Softw. Eng.: Softw. Eng. Pract. (ICSE-SEIP)*, Lisbon, Portugal, 2024, pp. 381–391. doi: 10.1145/3639477.3639753.

[4] S. Akhbarifar, H. H. S. Javadi, A. M. Rahmani, and M. Hosseinzadeh, "A secure remote health monitoring model for early disease diagnosis in cloud-based IoT environment," *Pers. Ubiquit. Comput.*, vol. 27, no. 3, pp. 697–713, 2023. doi: 10.1007/s00779-020-01475-3.

[5] A. Sundas *et al.*, "Smart patient monitoring and recommendation (SPMR) using cloud analytics and deep learning," *IEEE Access*, vol. 12, pp. 54238–54255, 2024. doi: 10.1109/ACCESS.2024.3383533.

[6] A. Bonci, E. Caizer, M. C. Giannini, F. Giuggioloni, and M. Prist, "Ultra Wide Band communication for condition-based monitoring, a bridge between edge and cloud computing," *Procedia Comput. Sci.*, vol. 217, pp. 1670–1677, 2023. doi: 10.1016/j.procs.2022.12.367.

[7] A. Ruiz-Zafra *et al.*, "NeoCam: An edge-cloud platform for non-invasive real-time monitoring in neonatal intensive care units," *IEEE J. Biomed. Health Inform.*, vol. 27, no. 6, pp. 2614–2624, 2023. doi: 10.1109/JBHI.2023.3240245.

[8] N. Soveizi, F. Turkmen, and D. Karastoyanova, "Security and privacy concerns in cloud-based scientific and business workflows: A systematic review," *Future Gener. Comput. Syst.*, vol. 148, no. 1343, pp. 184–200, 2023. doi: 10.1016/j.future.2023.05.015.

[9] Z. Xu, Y. Gong, Y. Zhou, Q. Bao, and W. Qian, "Enhancing kubernetes automated scheduling with deep learning and reinforcement techniques for large-scale cloud computing optimization," 2024, *arXiv: 2403.07905*.

[10] P. Rajadurai, "Machine learning-based secure cloud-IoT monitoring system for wireless communications," *DS J. Artif. Intel. Robot.*, vol. 1, no. 1, pp. 37–43, 2023. doi: 10.59232/AIR-V1I1P104.

[11] S. Wang, Z. Liu, Q. Sun, H. Zou, and F. Yang, "Towards an accurate evaluation of quality of cloud service in service-oriented cloud computing," *J. Intell. Manuf.*, vol. 25, no. 2, pp. 283–291, 2014. doi: 10.1007/s10845-012-0661-6.

[12] L. Qu, Y. Wang, M. A. Orgun, L. Liu, H. Liu and A. Bouguettaya, "CCCloud: Context-aware and credible cloud service selection based on subjective assessment and objective assessment," *IEEE Trans. Serv. Comput.*, vol. 8, no. 3, pp. 369–383, 2015. doi: 10.1109/TSC.2015.2413111.

[13] L. Xiao, G. Huang, and G. Zhang, "Improved assessment model for candidate design schemes with an interval rough integrated cloud model under uncertain group environment," *Eng. Appl. Artif. Intell.*, vol. 104, 2021, Art. no. 104352. doi: 10.1016/j.engappai.2021.104352.

[14] A. Sen and S. Madria, "Analysis of a cloud migration framework for offline risk assessment of cloud service providers," *Softw.: Pract. Exper.*, vol. 50, no. 6, pp. 998–1021, 2020. doi: 10.1002/spe.2809.

[15] Q. Liu *et al.*, "PQA-Net: Deep no reference point cloud quality assessment via multi-view projection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 12, pp. 4645–4660, 2021. doi: 10.1109/TCSVT.2021.3100282.

[16] Y. Liu, Q. Yang, Y. Xu, and L. Yang, "Point cloud quality assessment: Dataset construction and learning-based no-reference metric," *ACM Trans. Multimed. Comput., Commun. Appl.*, vol. 19, no. 80, pp. 1–26, 2023. doi: 10.1145/3603534.

[17] S. Wang, X. Wang, H. Gao, and J. Xiong, "Non-local geometry and color gradient aggregation graph model for no-reference point cloud quality assessment," in *Proc. 31st ACM Int. Conf. Multimed.*, 2023, pp. 6803–6810. doi: 10.1145/3581783.3612169.

[18] C. Zhang, G. Tian, A. M. Fathollahi-Fard, W. Wang, P. Wu and Z. Li, "Interval-valued intuitionistic uncertain linguistic cloud Petri net and its application to risk assessment for subway fire accident," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 1, pp. 163–177, 2022. doi: 10.1109/TASE.2020.3014907.

[19] A. Sen and S. Madria, "Application design phase risk assessment framework using cloud security domains," *J. Inf. Secur. Appl.*, vol. 55, 2020, Art. no. 102617. doi: 10.1016/j.jisa.2020.102617.

[20] F. K. Parast, C. Sindhav, S. Nikam, H. I. Yekta, K. B. Kent and S. Hakak, "Cloud computing security: A survey of service-based models," *Comput. Secur.*, vol. 114, no. 1, 2022, Art. no. 102580. doi: 10.1016/j.cose.2021.102580.