Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

File processing security detection in multi-cloud environments: a process mining approach

Xiaolu Zhang[1], Lei Cui[1], Wuqiang Shen[1], Jijun Zeng[1], Li Du[2], Haoyang He[2] and Long Cheng[1,2*]

Abstract

Cloud computing has gained popularity in recent years, but with its rise comes concerns about data security. Unauthorized access and attacks on cloud-based data, applications, and infrastructure are major challenges that must be addressed. While machine learning algorithms have improved intrusion detection systems in cloud data security, they often fail to consider the entire life cycle of file processing, making it difficult to detect certain issues, especially insider attacks. To address these limitations, this paper proposes a novel approach to analyzing data file processing in multi-cloud environments using process mining. By generating a complete file processing event log from a multi-cloud environment, the proposed approach enables detection from both control flow and performance perspectives, providing a deeper understanding of the underlying file processing in its full life cycle. Through our case study, we demonstrate the power and capabilities of process mining for file security detection and showcase its ability to provide further insights into file security in multi-cloud environments.

Keywords Cloud file security, Process mining, Intrusion dection, Process analysis

Introduction

Cloud computing has transformed the way data is stored, processed, and shared across organizations [1]. However, the growing popularity of cloud computing has raised serious concerns about data security. Unauthorized access and attacks on cloud-based data, applications, and infrastructure have been identified as the most significant challenge. Various attacks have been reported in recent years, including virtual machine (VM) escape attacks, distributed denial of service (DDoS) attacks, zero-day attacks, advanced phishing campaigns, and attacks on containers and cloud services [2].

To protect against these attacks, cloud computing security is essential, and it includes policies and procedures to prevent unauthorized access, data leakage, data alteration, software vulnerabilities, SQL injection, cross-site scripting, and flooding attacks. In recent years, the need for effective security measures to protect sensitive data stored in the cloud has become increasingly important, leading to the emergence of Intrusion Detection Systems (IDS) as a key tool in cloud data security [3]. Specifically, IDS are software or hardware-based systems that monitor network or system activities and detect unauthorized or malicious behavior. They can help cloud service providers and their clients prevent attacks, protect sensitive data, and ensure the availability of their services.

As cloud computing continues to grow and evolve, the use of IDS in cloud data security is becoming increasingly important in mitigating cyber threats and ensuring the privacy and security of sensitive data. Generally, there are two primary types of IDS: host-based IDS (HIDS) and network-based IDS (NIDS). HIDS monitors the

*Correspondence:
Long Cheng
lcheng@ncepu.edu.cn
[1] Joint Laboratory on Cyberspace Security, China Southern Power Grid, Guangzhou, China
[2] School of Control and Computer Engineering, North China Electric Power University, Beijing, China

operating system and runs on an individual machine. It tracks system calls, important files, and applications for any internal changes made by insiders and can inform the abnormal behavior, such as the modification or deletion of critical files, thereby preventing potential security breaches [4]. On the other hand, NIDS aims to monitor both internal and external cloud networks, and seeks to detect malicious activities or unusual behavior over the entire network, and provide a more comprehensive view of network security.

Although current research, such as machine learning algorithms, has significantly improved the capabilities of IDS, cloud data security still faces various challenges, particularly in multi-cloud environments. Machine learning algorithms can help IDS identify complex attacks and malicious activities that were previously difficult to detect [5]. However, the effectiveness of these algorithms can be limited by the quality of the training data and the ability to identify new and unknown threats. In addition, multi-cloud environments pose a unique challenge for cloud data security, as they involve multiple cloud service providers and require the coordination of various security protocols. In such environments, the complexity of managing and securing data across multiple clouds can create vulnerabilities that can be exploited by attackers. Therefore, cloud service providers and their clients must continue to develop and implement effective security measures to protect sensitive data in multi-cloud environments.

Process mining has been identified as a promising research direction for constructing behavioral or workflow models from event logs [6]. It enables a deeper understanding of underlying processes and facilitates the identification of anomalies and potential security threats [7]. Despite process mining being considered in security for some time [8], its application in cloud data security remains largely unexplored. This paper aims to explore the application of process mining in detecting data security threats in multi-cloud environments by tracking the full lifecycle of data files. Specifically, our approach aims to generate an event log of data file processing in multi-cloud environments and use process analysis to detect intrusion attempts. For instance, by detecting anomalous execution paths or file usage patterns at the level of processes, we can identify and prevent security attacks.

To the best of our knowledge, no current works have considered file processing security detection in a multi-cloud environment using process mining. Therefore, our approach presents a unique contribution to the field. Our approach provides an effective security mechanism for some types of attacks, including insider attacks in multi-clouds, which are notoriously difficult to detect. Additionally, our approach can be used as an additional line of defense, complementing existing security measures such as intrusion detection systems and firewalls, to provide a comprehensive and robust security solution for multi-cloud environments.

In general, the main contributions of this paper are summarized as follows:

- We propose a novel approach for analyzing data file processing in a multi-cloud environment using process mining. Our approach can construct an accurate event log that reflects the actual data file processing activities in a multi-cloud environment based on distributed events and messages logged in cloud systems.
- With the generated log, we perform a detailed analysis on the control and property level to identify potential security threats or anomalies in the data file processing process. This analysis allows us to better understand the behavior of the system and detect any malicious activity, providing enhanced security for multi-cloud environments.
- To evaluate the effectiveness of our approach, we conduct a case study on a multi-cloud dataset. The results show that our approach can effectively detect intrusion attempts, and it provides a unique and effective solution for data security detection in multi-cloud environments, which is becoming increasingly important in mitigating cyber threats and ensuring the security of sensitive data.

The rest of the paper is organized as follows. Related work section provides an overview of related work. Preliminaries section describes the proposed methodology. The proposed approach section presents the experimental evaluation. Our case study section discusses the results and limitations of the proposed approach. Finally, Conclusion section concludes the paper and outlines future research directions.

Related work

Cloud data security

Cloud computing has emerged as a popular computing paradigm, providing several advantages to organizations in terms of scalability, cost-efficiency, and flexibility [9, 10]. However, this paradigm also poses unique security challenges due to its distributed nature and resource sharing among multiple clients. Consequently, cloud security has become a vital topic of research in recent years, with a focus on data security, network security, and software security. To address these challenges, researchers have proposed various methods, including collaborative network security management systems. For example,

the work [11] proposes a practical collaborative network security management system to solve Internet security problems. Similarly, the research [12] presents a collaborative network security prototype system, vCNSMS, designed for a multi-tenant data center to protect against potential network attacks.

The security of cloud computing is a critical issue, and among the various challenges, cloud data security has gained significant attention due to the crucial role of data in cloud computing and its increasing adoption in various domains. The storage and processing of sensitive data on the cloud raise concerns about the confidentiality, integrity, and availability of data. As a result, several security measures have been developed, including encryption, access control, and intrusion detection systems (IDS). For instance, an efficient and secure access control model that employs attribute-based encryption (ABE), a distributed hash table (DHT) network, and identity-based timed-release encryption (IDTRE) has been proposed [13]. Similarly, a secure industrial data access control scheme for cloud-assisted Industrial Internet of Things (IIoT) that enables participants to enforce fine-grained access control policies for their IoT data via ciphertext policy-attribute-based encryption (CP-ABE) scheme has been designed [14]. To further strengthen cloud security, researchers have also developed intrusion detection systems (IDS) to detect and mitigate potential threats. For example, a framework for combating Distributed Denial of Service (DDoS) attacks in the cloud using an IDS has been proposed [15]. Similarly, an IDS for system security and protection, which performs real-time introspection of system events and analyzes them to detect potential threats, has been developed [16].

Recent research has highlighted the importance of enhancing the effectiveness and efficiency of intrusion detection systems (IDS) for cloud data security. One promising approach involves the use of machine learning algorithms to improve the accuracy and speed of anomaly detection in cloud environments. For instance, in a recent study [17], deep neural networks were utilized to develop a machine learning-based IDS capable of detecting and preventing both inside and outside attacks in cloud computing systems. In addition to IDS, researchers have also explored the use of blockchain technology to improve the security and privacy of cloud data storage and sharing. Blockchain's decentralized and immutable ledger can offer enhanced security and privacy for cloud data. One study [18] proposed a novel business model that employs consortium blockchain to secure and facilitate the sharing of data across multiple clouds, guaranteeing that the sharing is secure and trustworthy.

Generally, cloud data security remains a critical issue, and recent research has focused on developing more effective and efficient security measures to address the challenges of securing sensitive data in cloud computing environments. The use of machine learning, process mining, and blockchain technology has shown great promise in enhancing cloud data security and privacy. In this paper, our approach utilizes process mining for data security detection, which contributes significantly to research in the field of cloud data security.

Process mining for security

Process mining is a rapidly growing and promising area of research that focuses on understanding processes and capturing important information in the actual execution process [19, 20]. The goal of process mining is to improve operational processes through the systematic use of event data. This technique has been extensively studied and applied in many domains, such as healthcare, finance, and industry. For example, the work [21] demonstrates the application of process mining to analyze the bottlenecks in the maintenance of wind turbines within the energy system domain.

In recent years, there has been a growing interest in utilizing process mining to address security problems. Its ability to discover processes and check the conformance of running processes makes it a promising tool for security investigations compared to many existing methods. Specifically, process mining for security mainly focuses on anomaly detection, intrusion detection, attack detection, and insider threat detection [22]. For instance, the work [23] proposes a robust uncertain Business Process Management System (BPMS) architecture for accurately detecting anomalies compared to other algorithms. Another study [24] proposes a novel solution that utilizes process mining techniques to help operators identify attacks in IoT systems. Furthermore, many studies have combined process mining with other methods to address the aforementioned security issues. For example, a study [25] uses a clustering-based method paired with process mining to model the process in a more general format and disregard the system or protocol of particular IoT devices. For file processing processes, some works have used it for security analysis. For example, the work [26] proposes a software package that enables file processing information collection and applies it to information systems to achieve information security incident investigation.

Process mining is also an applicable and extensively studied technique in addressing security challenges in cloud computing. One study [27] proposed an error detection method using process mining techniques aimed at sporadic operations, and presented a specific use case in a cloud environment. In comparison, this paper focuses on the use of process mining for data

security detection in a multi-cloud environment, which presents a stronger challenge to cloud data security due to its complex structure and the involvement of multiple cloud service providers along with various security rules. Our approach addresses this challenge by utilizing process mining to identify security issues in the multi-cloud environment, which can help strengthen cloud data security in this complex setting.

Preliminaries

In this section, we will present some key concepts of process mining. To maintain consistency, we mainly adhere to some established notations provided in the work [28].

Definition 1 (Event). In the context of data file security in cloud, an event is the occurrence of an activity about file processing being executed. Specifically, we define an event as a 4-tuple $e = (act, cid, t, p)$, where *act* represents the name of the activity that was executed, *cid* is the unique identifier of the process instance to which the event belongs, t denotes the time at which the event occurred, which could be the start or end time of the activity, and p is the attribute associated with the event, such as the security level and cloud location of the file.

Definition 2 (Trace). A trace is a finite and non-empty sequence of events $\sigma = \langle e_1, e_2, ..., e_{|\sigma|} \rangle$ such as that for $\forall i, j \in [1, |\sigma|]$, if $i < j$, then there are $\sigma_i \neq \sigma_j$, $\sigma_i = \sigma_i$, $e_i.cid = e_j.cid$, and $e_i.t < e_j.t$. In another words, all events in the same trace are ordered by their timestamps.

Definition 3 (Case). A case is a completed trace, it can be expressed as a 2-tuple $c = (cid, \sigma)$, where *cid* is the case identifier, and σ is the event trace of the case. σ_i represents the i-th event in trace σ, $\sigma_i.cid = cid$, the start event of case is σ_1, and the end event is $\sigma_{|\sigma|}$.

Definition 4 (Event log). An event log is a set of the completed traces of cases, which records the actual execution for the processing of a file. It can be expressed as $L = \{\sigma_1, \sigma_2, ..., \sigma_{|L|}\}$, where $|L|$ represents the length of the event log.

Definition 5 (Anomalous Trace). Let $P(L(p))$ be a function that measures the pattern of a set of traces with the same properties p in terms of control flow or performance behavior. Then, an anomalous trace σ satisfies: $\sigma \in L(p)$ such that $|P(\sigma) - P(L(p))| > \theta$, where θ is a threshold defining the level of deviation that is considered anomalous. In our work, we use the deviation to find potential security threats or anomalies in a data file processing process.

Table 1 gives an example of the fragment of an event log, where each row corresponds to a single event, and each column represents an attribute of the event. Specifically, the *Case id* column indicates the identifier of the case in file processing, while the *Activity* and *Timestamp* columns denote the name of the activity and its execution timestamp, respectively. For instance, in *case*1, the execution trace $\langle create, read, modify, read \rangle$ begins with event *create*, which occurred at 2022-12-30 11:02, and ends when event *read* is completed.

The proposed approach

This section outlines our proposed approach for detecting file processing security issues in cloud computing. Figure 1 provides an overview of our approach. The cloud system(s) will record file processing events, and an event log will be generated based on the collected events from each system. This event log will then be used for security detection. Our approach includes log generation, control flow detection, and performance mining.

Inter-cloud file processing log generation

In a multi-cloud environment, as depicted in Fig. 2, each cloud platform has an independent file processing process, referred to as *internal processing*. Each internal processing consists of internal file operations, which are denoted as *internal activity nodes*. Communication between different cloud platforms occurs through communication requests, which are referred to as *communication nodes*. Communication nodes can be classified as either file sending nodes or receiving nodes. Therefore, an internal processing flow can be defined as $W_i = (A_i, E_i)$, where W_i denotes the data file processing process in the i^{th} cloud platform, A_i is the set of all internal activity nodes in the process, and E_i denotes the set of all edges in the process. Each internal activity node a_k in the process is represented as a quadruplet:

Table 1 An example of event logs

Case id	Activity	Timestamp	Sec. Level	Cloud Loc.
1	create	2022-12-30 11:02	3	B
	read	2022-12-30 11:06	3	B
	modify	2022-12-30 11:12	3	B
	read	2022-12-30 11:18	3	B
2	create	2022-12-30 16:10	5	A
	modify	2022-12-30 16:14	5	A
	move	2022-12-30 16:26	5	A
	read	2022-12-30 16:36	5	B
	copy	2022-12-30 16:40	5	B
	...	...	...	...

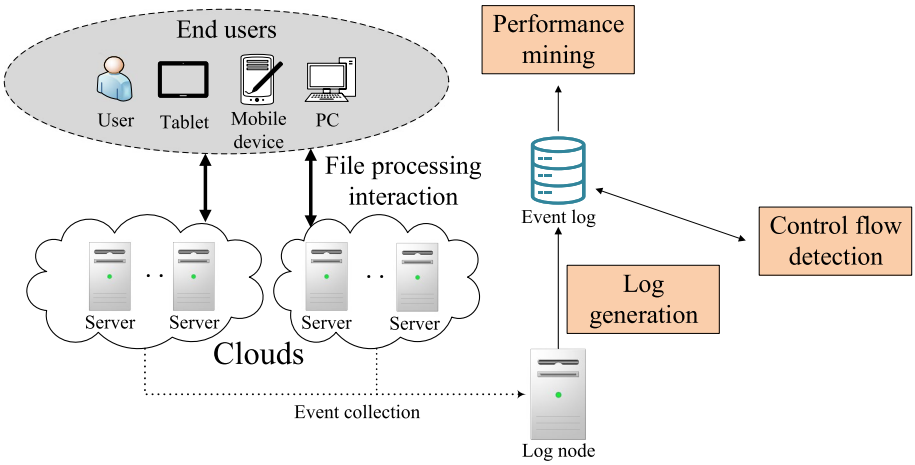

Fig. 1 The general framework of the proposed process mining approach for file processing security detection in cloud

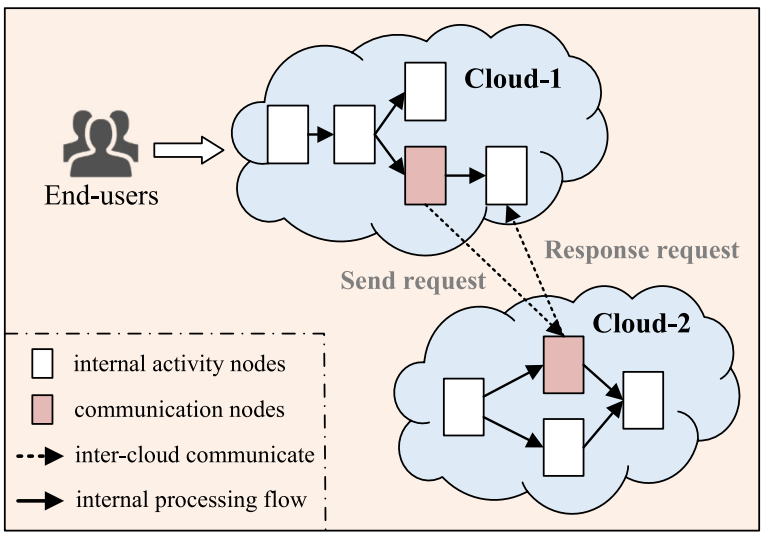

Fig. 2 The data file processing in a multi-cloud environment

$a_k = (name_k, time_k, person_k, s_k)$, where $name_k$ denotes the name of activity k, $time_k$ denotes the operation time, $person_k$ denotes the ID of the operator, and s_k denotes the security level of the operation.

In addition, when analyzing the communication process between inter-cloud platforms, we consider two data file processing processes, W_i and W_j, in two different cloud platforms. The messages exchanged between these processes can be represented as ordered pairs $(M_{send}, M_{receive})$, where $M_{send} \in W_i$ and $M_{receive} \in W_j$ denote the activities of sending and receiving requests, respectively. Each communication node is represented as a six-tuple: $M_c = (name_c, time_c, person_c, s_c, cid_c, attr_c)$, where cid_c denotes the ID of the local cloud platform and $attr_c$ indicates whether the node is a sending node or a receiving node.

During data file processing in the cloud platform, each local operation $A_i^{'}$ related to the file will form a process according to the time sequence and be recorded in the local cloud event log L, where $A_i^{'} \in A_i$. However, the distributed nature of cloud computing presents a significant challenge to the process mining of data files and poses a risk to the security of data files across cloud platforms. Specifically, in a cloud computing environment, data files are not stored or processed in a single place, but instead, they are spread out across multiple servers, possibly in different geographical locations. When data is distributed, it becomes more difficult to collect all the

necessary information as it can complicate the order and timing of events. To address this issue, we propose using a combination of local logs and communication logs for process analysis. Specifically, we assume that each cloud platform needs to record inter-cloud platform communication logs CL in addition to local logs. For cloud platform W_i, all its communication activities are recorded in the communication logs. Given a workflow W_i in a cloud platform, the communication log CLi of the platform is composed of all its message exchange pairs, i.e., $CLi = \{(Msend, Mreceive) | M_{send} \text{ or } M_{receive} \in A_i\}$.

We can obtain the actual data file processing flow in the multi-cloud environment by combining the local log L_i and communication log CLi. Let $Lall = L_i \cup CL_i$, and then analyze the causal relationships between the processes and remove any redundant items. Specifically, the internal data file processing process is recorded in the distributed local log L_i, while the inter-cloud communication process is recorded in the communication log CL_i. After combining the logs, we analyze the causal relationships between the processes to remove any redundant items and obtain a complete data file processing event log. This approach enables us to leverage process mining techniques to detect any security issues in the data file processing flow and improve the overall security and privacy of data files in multi-cloud environments.

Process analysis for intrusion detection

Control flow of file processing

Detecting control flow problems in data file processing is critical for ensuring the efficiency, reliability, and security of files in multi-clouds. Control flow problems such as deadlocks and loops can cause processes to stall or become unresponsive and result in security vulnerabilities, as attackers may exploit control flow issues to gain unauthorized access to sensitive data or manipulate the process flow to their advantage. Detecting control flow problems early on can prevent these issues from escalating and help organizations maintain the integrity and availability of their data processing systems.

Conformance checking is a process mining technique used to compare the expected behavior of a business process with its actual execution [29]. By comparing the discovered process model, which represents the main behaviours of file processing in multi-cloud, to the collected event logs, conformance checking can identify discrepancies or deviations from the expected behaviour, which may indicate control flow problems in data file processing. To perform conformance checking, there are different techniques that can be used. One commonly used approach is the optimal alignment algorithm, which aims to find the best alignment between a process model and an event log by minimizing the distance between them. The goal is to identify the best fit between the actual process behavior captured in the event log and the expected behavior represented in the process model. This approach can be used to identify deviations, bottlenecks, and potential improvements in the process, as well as to evaluate the quality of the process model. More detailed information about the approach and its implementations can be found in the referenced book [28].

Multi-dimensional analysis of file processing

Multi-dimensional process analysis is a process mining technique that allows for the analysis of complex business processes with multiple dimensions [30]. It enables the analysis of the process from multiple perspectives, such as time, resource, and data dimensions. In multi-cloud environments, multi-dimensional process analysis can be used for anomaly detection of file processing activities by analyzing the event logs from different clouds and identifying and flagging unusual or unexpected behavior in a system. For example, in multi-cloud environment data file processing, it can be used to detect unusual patterns of data transfer or access that may indicate intrusion or security threats.

Out of all the available multi-dimensional process analysis technologies, we have selected the performance spectrum [31] as one of the latest techniques for identifying unusual patterns from a performance perspective. This approach offers a fresh perspective on measuring and analyzing process performance, encompassing efficiency, effectiveness, and quality. Rather than relying on a process model, the performance spectrum enables fine-grained performance analysis by combining all observed flows between two process steps based on their performance over time [32]. This approach offers a more accurate and detailed understanding of process performance, making it a valuable tool for identifying anomalies and detecting potential security threats in multi-cloud environments. In addition, it is important to highlight that we can leverage various other properties associated with events in an event log to facilitate multi-dimensional process analysis. For instance, factors like the security level and cloud location of a file used in our case study in the following section can be utilized if required.

Generally, the generation of the performance spectrum for file processing is presented in Algorithm 1. This algorithm takes as input a set of process instances $\mathbf{P}$ and outputs the performance spectrum matrix $\mathbf{S}$. For each process instance p_i, the algorithm iterates over every pair of process steps (a_j, a_k) and calculates the time difference $t_{jk}(p_i)$ between the completion of a_j and the start of a_k in p_i. The algorithm then updates the corresponding entry s_{jk} in $\mathbf{S}$ using a function $f(t_{jk}(p_i))$ that maps the time difference to a performance metric. Once all process instances have been processed, the resulting performance spectrum matrix $\mathbf{S}$ is returned.

Input: A set of process instances for file processing $\mathbf{P}$
Output: Performance spectrum matrix $\mathbf{S}$
Initialize an empty performance spectrum matrix $\mathbf{S}$ with size $n \times m$, where n is the number of process steps and m is the number of performance metrics; **foreach** *process instance* $p_i \in \mathbf{P}$ **do**
  foreach *pair of process steps* (a_j, a_k) **do**
    Calculate the time difference $t_{jk}(p_i)$ between the completion of a_j and the start of a_k in p_i; Update the corresponding entry s_{jk} in $\mathbf{S}$ using a function $f(t_{jk}(p_i))$ that maps the time difference to a performance metric;
  end
end
return S

Algorithm 1 Performance spectrum generation for file processing

With the above algorithm, we then can use the performance spectrum for anomaly detection in multi-cloud environments is to establish a baseline for the expected performance of the file processing activities. This can be done by analyzing historical event logs and establishing a set of performance metrics that represent the expected behavior of the process. Once a baseline has been established, it is possible to monitor the performance metrics in real-time and compare them to the expected values. For instance, if the performance metrics deviate from the expected values, this may indicate an anomaly in the file processing activities.

Our case study

Our goal is to investigate the potential of process mining to identify data security threats in multi-cloud environments, by comprehensively tracking the lifecycle of data files. To achieve this, our approach employs process analysis techniques, which enable us to detect security issues related to the processing of data files in multi-cloud environments. In this study, we have taken the following three steps based on the collected event log: data preprocessing, control flow analysis, and multi-dimensional analysis. By implementing these steps, we aim to gain a deeper understanding of the security risks that exist in multi-cloud environments, and to develop effective strategies for mitigating these risks.

Data pre-processing

We first perform the pre-processing over a sample of an event log generated from a multi-cloud environment with the following operations:

1 Remove incomplete cases, as well as cases that have been recorded not from their beginning.
2 Extract relevant data attributes, and determine *case id*, *activity*, *timestamps*, we sort the events of a trace based on the *start timestamp*.
3 Mark some rare values (in less than 10 cases) as other for categorical variables with many possible values.
4 Check if there are any data attributes that are constant across all cases and events, or cross-correlated with other attributes and discard them.
5 Perform some basic feature engineering, such as extracting weekday, hour, and duration since the previous event in the given case, and elapsed time since the beginning of the case using event timestamps from the log.

Table 2 presents the details of the logs. There are 8 unique activities in the dataset, and the operating cloud platform location information of file processing is recorded in the *Cloud* attribute.

Control flow analysis

To assess how closely the handling of files in multi-cloud environments adheres to the expected execution flow, we conducted an alignment-based conformance check of the

Table 2 Statistics of the dataset

Event Log	File_Processing
Cases	1000
Trace variants	839
Events	14025
Events per case (mean)	14.025
Median case duration (hours)	1.71
Mean case duration (hours)	2.41
Activities	8
Cloud location	Cloud

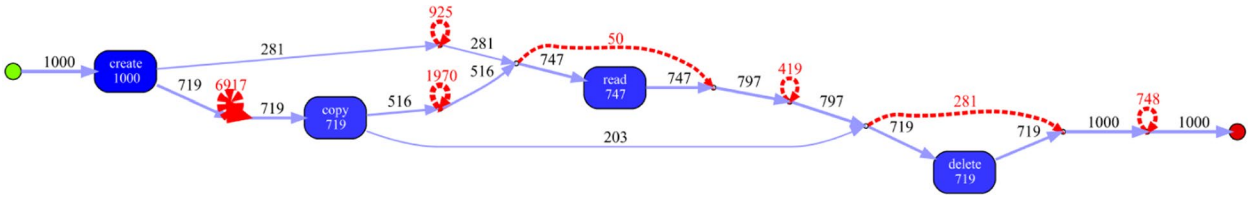

Fig. 3 The DFM annotated with conformance and frequency information

event log against a standard model generated by adjusting the parameters in the DFM plugin [33]. This involved aligning a trace with a process model to identify any deviations, which can be categorized into three types of moves:

1 Log move: When an event in the log is not supported by the model.
2 Model move: When an expected behavior in the model is not observed in the log.
3 Synchronized move: When there is a match between a recorded behavior and a defined activity in the model.

As shown in Fig. 3, our analysis revealed several deviations from the expected execution flow, represented by red dotted lines. Model moves are depicted as lines crossing activities in the process model, while log moves are shown as loops over nodes. Our analysis revealed that the activity *delete* was skipped 281 times during the given period, and there were significant deviations in the execution of *create*, indicating a need for further investigation to identify the underlying causes.

Overall, our analysis using alignment-based conformance checking provides valuable insights into the handling of files in multi-cloud environments, highlighting areas where improvements can be made to optimize performance and ensure that operations are executed as expected.

Multi-dimensional analysis

To conduct a detailed performance analysis of the data file processing process in multi-cloud environments, we utilized the start and end timestamps of each event to identify potential bottlenecks or behaviors that may affect the overall performance of the process. The throughput of the model discovered for data file processing in multi-cloud environments is shown in Fig. 4. In this chart, the execution time of each activity is annotated, with darker colors indicating longer processing times. Our analysis revealed that the activity *move* takes around 11 minutes. Based on our experience and the size of files being transferred, we can determine whether such a data transfer time is normal or not. This information can be used to identify potential issues and areas for optimization in the data file processing process in multi-cloud environments.

While the techniques described earlier provide valuable insights into the performance of the data file processing process in multi-cloud environments, they only offer a coarse-grained analysis by aggregating all event data for each process step. To enable more accurate and detailed process performance analysis, we utilized the performance spectrum as a simple model that maps all observed flows between two process steps based on their performance over time. As shown in Fig. 5, the performance spectrum provides a detailed view of the process performance, with many vertical and inclined lines corresponding to multiple observations distributed over time. Each line is colored based on the duration, and

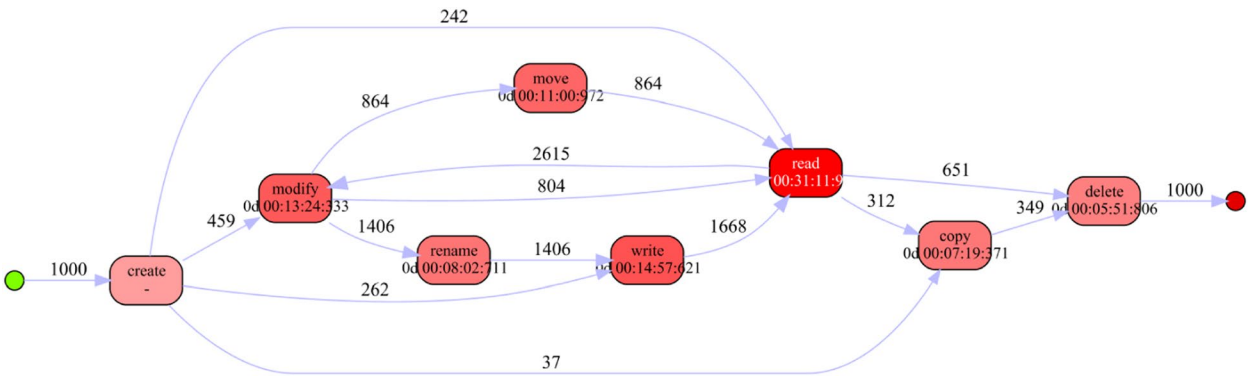

Fig. 4 The performance analysis for data file processing in multi-cloud environment

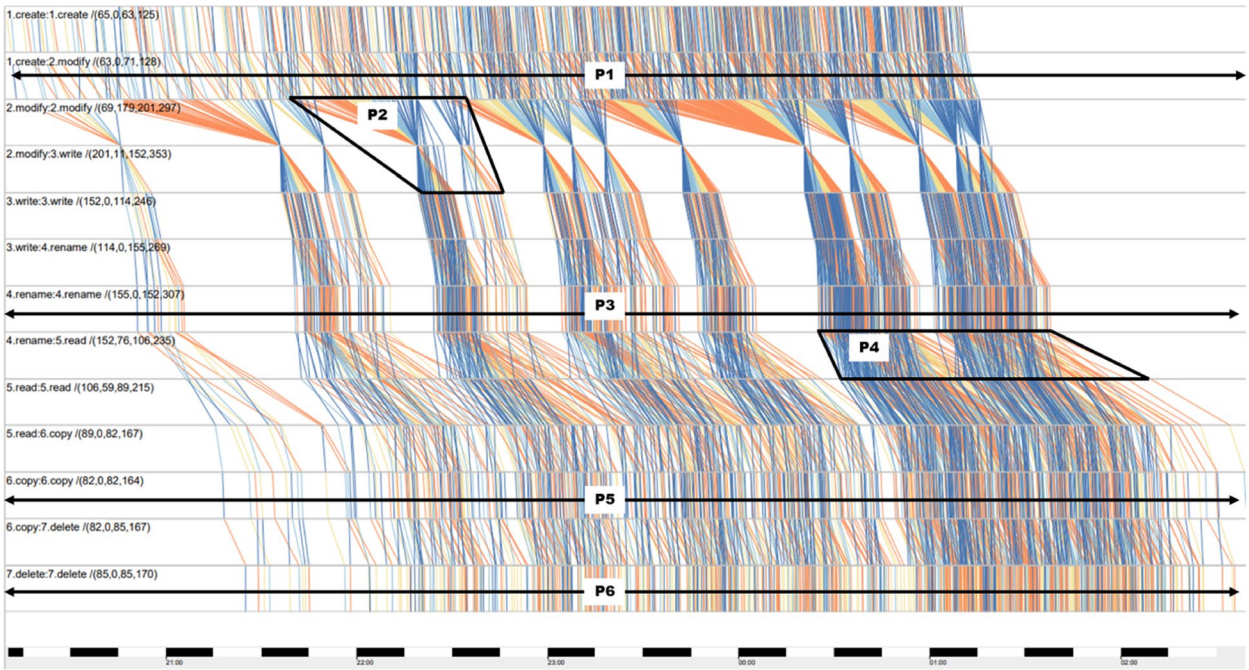

Fig. 5 The performance spectrum for data file processing in multi-cloud environment

non-crossing lines show a strict FIFO order. Identical inclinations indicate a constant waiting time for all cases. Variation in the density of the lines (and in the height of the bars of the aggregated performance spectrum) reveals continuous, varying workload throughout the entire log, with the color of the lines indicating performance.

From the results, we can reveal that the *create:modify* step of P1 globally contains many traces of variable duration, which are continuously distributed over time and can overtake each other. We observed that operators modify files at various speeds, leading to significant variability in performance. In addition, the composite pattern P2 consists of two different performance variants, with the "sand clock" pattern indicating cases accumulated over a period of approximately 45 minutes. For this case, we can hypothesize that the file operation necessitates system approval, or it may be under attack. Furthermore, the exceptionally brief duration of certain operations within pattern P2 raises valid concerns about their legitimacy. Such rapid activity could potentially pose security risks, including unauthorized alterations and malevolent tampering. Patterns P3 *rename:rename*, P5 *copy:copy*, and P6 *delete:delete* show instant processing of cases. However, the visualization of P4 *rename:read* reveals a large gap in the time between renaming and reading files in the cases, highlighting the need for further investigation to identify the underlying causes with possible security issues.

Conclusion

This paper proposes a novel approach for analyzing data file processing in multi-cloud environments using process mining. By generating a complete file processing event log and performing control flow and performance analysis, the proposed method provides a more comprehensive solution for detecting and mitigating security threats in multi-cloud environments, with addressing the limitations of existing approaches that have not considered the full life cycle of file processing. Through our case study, we demonstrate the power and capabilities of process mining for file security detection, especially in complex multi-cloud environments. As future work, this approach can be extended to address additional challenges in multi-cloud environments, such as ensuring data privacy and confidentiality. Further investigation can also be done to explore the potential of combining process mining with other techniques, such as machine learning, to enhance the overall effectiveness of the approach.

Acknowledgements
Not applicable.

Authors' contributions
Xiaolu Zhang: Conceptualization, Writing - original draft, Writing - review & editing. Lei Cui: Methodology, Writing - review & editing. Wuqing Shen: Writing - review & editing. Jijun Zeng: Writing - review & editing. Li Du: Methodology, Writing - original draft, Writing - review & editing. Haoyang He: Writing - review & editing. Long Cheng: Conceptualization, Methodology, Writing - original draft, Writing - review & editing.

Funding
This work was supported by the Open Project Program of the Joint Laboratory on Cyberspace Security, China Southern Power Grid (No. CSS2022KF01).

Availability of data and materials
The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 13 April 2023 Accepted: 14 June 2023
Published online: 06 July 2023

References

1. Liu J, Shen H, Chi H, Narman HS, Yang Y, Cheng L, Chung W (2020) A low-cost multi-failure resilient replication scheme for high-data availability in cloud storage. IEEE/ACM Trans Netw 29(4):1436–1451
2. Lata S, Singh D (2022) Intrusion detection system in cloud environment: Literature survey & future research directions. Int J Inf Manag Data Insights 2(2):100134
3. Wang W, Du X, Shan D, Qin R, Wang N (2020) Cloud intrusion detection method based on stacked contractive auto-encoder and support vector machine. IEEE Trans Cloud Comput 10(3):1634–1646
4. Park D, Kim S, Kwon H, Shin D, Shin D (2021) Host-based intrusion detection model using siamese network. IEEE Access 9:76614–76623
5. Li J, Tong X, Liu J, Cheng L (2023) An efficient federated learning system for network intrusion detection. IEEE Syst J. 17(2):2455-64
6. Cheng L, van Dongen BF, van der Aalst WM (2020) Scalable discovery of hybrid process models in a cloud computing environment. IEEE Trans Serv Comput 13(2):368–380
7. Liu C, Zeng Q, Cheng L, Duan H, Zhou M, Cheng J (2020) Privacy-preserving behavioral correctness verification of cross-organizational workflow with task synchronization patterns. IEEE Trans Autom Sci Eng 18(3):1037–1048
8. Van der Aalst WM, de Medeiros AKA (2005) Process mining and security: Detecting anomalous process executions and checking process conformance. Electron Notes Theor Comput Sci 121:3–21
9. Cheng L, Kotoulas S (2018) Efficient skew handling for outer joins in a cloud computing environment. IEEE Trans Cloud Comput 6(2):558–571
10. Cheng L, Kalapgar A, Jain A, Wang Y, Qin Y, Li Y, Liu C (2022) Cost-aware real-time job scheduling for hybrid cloud using deep reinforcement learning. Neural Comput Appl 34(21):18579–18593
11. Chen Z, Han F, Cao J, Jiang X, Chen S (2013) Cloud computing-based forensic analysis for collaborative network security management system. Tsinghua Sci Technol 18(1):40–50
12. Chen Z, Dong W, Li H, Zhang P, Chen X, Cao J (2014) Collaborative network security in multi-tenant data center for cloud computing. Tsinghua Sci Technol 19(1):82–94
13. Namasudra S (2019) An improved attribute-based encryption technique towards the data security in cloud computing. Concurr Comput Pract Exp 31(3):e4364
14. Qi S, Lu Y, Wei W, Chen X (2020) Efficient data access control with fine-grained data protection in cloud-assisted iiot. IEEE Internet Things J 8(4):2886–2899
15. Nagar U, Nanda P, He X, Tan Z (2017) A framework for data security in cloud using collaborative intrusion detection scheme. In: Proceedings of the 10th International Conference on Security of Information and Networks. ACM, pp 188–193
16. Snehi J, Snehi M, Bhandari A, Baggan V, Ahuja R (2021) Introspecting intrusion detection systems in dealing with security concerns in cloud environment. In: 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART). IEEE, pp 345–349
17. Chiba Z, Abghour N, Moussaid K, Rida M et al (2019) Intelligent approach to build a deep neural network based ids for cloud environment using combination of machine learning algorithms. Comput Secur 86:291–317
18. Shen M, Duan J, Zhu L, Zhang J, Du X, Guizani M (2020) Blockchain-based incentives for secure and collaborative data sharing in multiple clouds. IEEE J Sel Areas Commun 38(6):1229–1241
19. Liu C, Cheng L, Zeng Q, Wen L (2022) Formal modeling and discovery of hierarchical business processes: A petri net-based approach. IEEE Trans Syst Man Cybern Syst 53(2):1003–14
20. Liu C, Li H, Zhang S, Cheng L, Zeng Q (2022) Cross-department collaborative healthcare process model discovery from event logs. IEEE Trans Autom Sci Eng
21. Du L, Cheng L, Liu C (2021) Process mining for wind turbine maintenance process analysis: A case study. In: IEEE 5th Conference on Energy Internet and Energy System Integration. IEEE, pp 3274–3278
22. Silalahi S, Yuhana UL, Ahmad T, Studiawan H (2022) A survey on process mining for security. In: 2022 International Seminar on Application for Technology of Information and Communication (iSemantic). pp 1–6. https://doi.org/10.1109/iSemantic55962.2022.9920473
23. Saraeian S, Shirazi B (2020) Process mining-based anomaly detection of additive manufacturing process activities using a game theory modeling approach. Comput Ind Eng 146:106584
24. Coltellese S, Maria Maggi F, Marrella A, Massarelli L, Querzoni L (2019) Triage of iot attacks through process mining. In: On the Move to Meaningful Internet Systems: OTM 2019 Conferences: Confederated International Conferences: CoopIS, ODBASE, C &TC 2019, Rhodes, Greece, October 21–25, 2019, Proceedings. Springer, pp 326–344
25. Hemmer A, Badonnel R, Chrisment I (2020) A process mining approach for supporting iot predictive security. In: NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium. IEEE, pp 1–9
26. Gaidamakin N, Gibilinda R, Sinadskiy N (2020) File operations information collecting software package used in the information security incidents investigation. In: 2020 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT). IEEE, pp 559–562
27. Yang H, Park M, Cho M, Song M, Kim S (2014) A system architecture for manufacturing process analysis based on big data and process mining techniques. In: 2014 IEEE International Conference on Big Data (Big Data). IEEE, pp 1024–1029
28. Van Der Aalst W (2016) Process mining: data science in action. Springer
29. Cheng L, Liu C, Zeng Q (2023) Optimal alignments between large event logs and process models over distributed systems: An approach based on Petri nets. Inf Sci 619:406–420
30. Bolt A, van der Aalst WM (2015) Multidimensional process mining using process cubes. In: Enterprise, Business-Process and Information Systems Modeling: 16th International Conference. Springer, pp 102–116
31. Fahland D (2022) Multi-dimensional process analysis. In: Proceedings of the 20th International Conference on Business Process Management, vol 13420. Springer, pp 27–33
32. Denisov V, Belkina E, Fahland D, van der Aalst WM (2018) The performance spectrum miner: Visual analytics for fine-grained performance analysis of processes. In: BPM (Dissertation/Demos/Industry). Springer, pp 96–100
33. Leemans S, Poppe E, Wynn M (2019) Directly follows-based process mining: A tool. In: Proceedings of the ICPM demo track 2019. IEEE, pp 9–12

Publisher's Note