MsFaaS: Microservices Forensics as a Service

Abubakr Shehata
Nile University

Heba Aslan (✉ haslan@nu.edu.eg)
Nile University

Research Article

Keywords: Digital forensics, forensics investigation, DFRWS, forensics-as-a-service, microservices, events reconstruction

Posted Date: January 25th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2511365/v1

License: ☑️ 🔐 This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License
Abstract

The rapid increase in using cloud services leads to the rise of unlimited cyber threats. Due to the distributed architect nature, services like Software as a Service (SaaS) are hard to be monitored. Consequently, in digital crimes, the investigator faces several forensics challenges. Mainly, in collecting data and presenting acceptable electronic evidence at the court. While most researchers went towards the external investigator’s responsibility, fewer researchers went to the Cloud Service Provider’s (CSP) responsibility. In literature, no complete framework that resolves all challenges and succeeds in presenting trusted evidence to the court has been suggested. In the present paper, we propose a framework named Microservices Forensics as a Service (MsFaaS) that suggests solutions to resolve open challenges. Since the CSP holds the crime scene and there is a global legal responsibility against digital crimes, MsFaaS relies on enforcing international law and putting the responsibility on both CSP and local authorities where the victim/attacker and CSP operate. We collect several data types and use event reconstruction methodology to build a comprehensive crime scene in both live and postmortem scenarios. In addition, MsFaaS provides a validation service against man-in-the-middle attacks microservices. It also triggers a response action to virtual machines incidents and external attacks; finally, the framework is helpful for auditing and recording the external and internal traffic inside the cloud environment. As a result, the service provided by the framework offers an evidence report that is certified by the local authority which can be accepted in a digital crime trial.

1. Introduction

Cloud providers have enormous services across nations and data centers worldwide. Multi-tenancy and nodes are distributed geographically in different locations across nations. Due to the heterogeneous architect for SaaS microservices, it is more vulnerable to attacks than the conventional monolithic architect. In addition, the firewalls and monitoring tools easily control the monolithic architect, since all its features pass through a single point. Therefore, all evidence is in a central location, and the forensics mission is durable. In contrast, the microservices environment has a mutable characteristic as follows. The components and logs are geographically distributed. It has multiple interfaces; each microservice (MS) could be deployed on a different virtual machine, container or directly hosted on a hypervisor node [1]; each MS could communicate to each other over a scalable, loosely coupled, and self-contained network [2]. Consequently, many security threats were generated, especially for microservices [3]; several forensics challenges prevented the Digital Forensics Investigator (DFI) from collecting valid evidence; the evidence presented at the trial has a broken Chain of Custody (CoC). As a result, the court rejects such digital evidence, since it is not certified by an authorized organization, and it could have been forged.

In literature, the work proposed to resolve the forensics challenges was created in three sectors. First, the National Institute of Standards and Technology (NIST) identified the challenges by creating a report named NISTIR 8006 [4] in June 2014 and updated in August 2020. The report classified the challenges under nine categories that prevented DFI from presenting solid evidence to the court. Second, the Digital Forensic Research Workshop (DFRWS) [5] defined six phases to determine the best practice cloud
forensics’ methodology. Third, some solutions in the literature were created to resolve the challenges mentioned in NIST’s report. Each researcher focused on resolving some challenges’ categories without presenting a complete solution to resolve all challenges. They used solutions like a central forensics’ server, external third-party interaction, and data encryption for security and integrity. They are concentrated only on data collected from the network traffic and files related to virtual machines ignoring the rest of the data types needed to build a full image of the crime scene. Other solutions used machine learning and blockchain to achieve data integrity but did not keep the CoC which is the core requirement for presenting trusted evidence. Accordingly, six out of nine challenges have been resolved, as follows: architect, first response, anti-forensics, analysis, training, and role management. On the other hand, data collection, standardization, and legality remained uncovered. The reason for the failure is that researchers ignored collecting essential data types such as firewall logs, and information about the microservices, dumping images from Virtual Machines (VM) or the container that holds the microservice. In addition, they ignored law enforcement. The law obligates governments (for example: the National Telecom Regulation and Authority (NTRA) of Egypt) and international bodies to protect and preserve data that supports investigations into digital crimes. For example, European Union (EU) General Data Protection Regulation (GDPR) compliance enforces personal data protection [6]. Many researchers built their model based on the external DFI approach, which involves limited resource accessibility and a lack of the required authority to implement the proposed solutions. Another example is when VMs’ logs records show the infection of malware while other data types, such as customer relation to this VM, are not available. On this occasion, failure to collect all related data obviously led to a broken CoC. Consequently, the current literature failed to provide a complete solution for presenting solid Electronic Evidence (EE) to the court.

In this paper, we propose the Microservices Forensics as a Service (MsFaaS) framework. The solution resolves the data collection, standardization, and legality challenges. It is built on collecting solid ideas from previous literature and adding solutions for the data collection, standardization, and legality challenges. MsFaaS starts with prerequisites as a preparation for the solution. It needs to be implemented and operated by the cloud provider. Hence, it collects all related information from several data sources such as network traffic, swap files [7], VM dump [8] or containers, firewall logs, microservices activities, and users’ registered data. Before the incident occurred, we used event reconstruction [9] methodology to build a comprehensive image of the crime scene in both live and post-mortem scenarios [10]. The solution flow is guided by the DFRWS methodology. It starts with data collection, segregation, adding extra attributes, normalization, examination, analysis, and presentation at the court. We kept the data integrity using blockchain [11], stored in hashed [12] format, and we added a third-party server [13] located in a governmental body to achieve a trust factor. We replicate the hashed values to the server at the local government authority which in turn provides a certified report verifying the complete CoC. Therefore, MsFaaS acts as a forensics and security tool. For forensics, the framework provides evidence with a complete CoC which is accepted in a court case. In addition, it is helpful in monitoring and recording external attacks or internal threats from inside the cloud environment. For security, it provides a validation method as a countermeasure against Man-In-The-Middle (MITM) attacks
on microservices. In addition, it triggers an incident response in case of VMs malware infection or hacking activities. It transforms the intangible distributed metadata from raw form into correlative pieces of tangible EE in the form of a certified report that could be presented at the court.

The main paper’s contributions are:

- MsFaaS framework resolves three uncovered NIST challenges: Data collection, standardization, and legality.
- MsFaaS framework provides the following services: Forensics-certified report, CSP auditing, and MSs security validation.
- MsFaaS framework enhances the analysis by ML technique.
- MsFaaS framework provides (eCommerce) validation service as a use case against MITM attack.

The rest of this paper is organized as follows: the background and literature review is detailed in Section 2. The framework that describes our proposed framework (MsFaaS) is detailed in Section 3. Finally, the paper concludes in Section 4.

2. Background And Literature Review

Due to the importance of cloud forensics worldwide, academic studies have been conducted. However, they covered only some of the challenges, and there was no complete solution has been achieved yet. In order to have an organized cloud forensics framework, NIST created a report to list cloud forensics challenges, organized conferences, and created phases to improve forensics methodology. In the following subsections, we illustrate the work done by NIST; we also enumerate some of the solutions proposed in the literature.

2.1 Background

In this subsection, we give a detailed description of the efforts done by NIST and DFRWS concerning the challenges facing a secure digital forensics investigation and the phases needed to complete it.

Challenges

Use paragraph headings as needed. As per the NIST report, cloud forensics challenges are classified into categories and subcategories. The revealed issues are technical, while legality and organizational issues come next. Technical challenges vary according to cloud computing services such as: Platform as a Service (PaaS), Infrastructure as a Service (IaaS), and SaaS [14]. The SaaS microservice is scalable and developed by different teams and platforms. It uses on-demand self-service, which gets its resource elasticized from a pool of resources. The legal and organizational issues caused by cloud distribution across national borders and legal jurisdictions. The main categories are presented in red text in Fig. 1, and the sub-categories are presented in green text to identify the details of the challenge.
Challenges’ details are described as follows: The Architecture is complicated due to the microservices concept and multi-tenancy. Investigators suffer from Data Collection, as it is in a pool of resources distributed on multiple nodes and the data traffic cannot be verified whether it is from a trusted microservice or generated by man-in-the-middle hackers. In addition, the data volatility nature of these artifacts represents another aspect of complexity, as each microservice or virtual machine is exposed to be deleted, including its logs and swap files (vRAM and vHD). Legal issues such as integrity, validity, and privacy prevent investigators from getting valuable data in postmortem scenarios. Moreover, the Service Level Agreement (SLA) does not guarantee to keep logs history after service deletion, which leads to a broken CoC at the trial. The variety of forensics tools and practices increases the standardization challenge. There are no standards that govern all CSPs or investigators’ forensics testing and validation. The Analysis is also a big challenge due to the need to correlate different file versions and formats with events after they were reconstructed in a metadata form. In addition, the collected data have different timestamps due to the worldwide different time synchronization. Anti-forensics concept is a set of techniques that are used specifically to prevent or mislead forensic analysis. Methods used to evade forensics tools include: obfuscation, malware, data hiding, or other techniques to compromise the integrity of evidence. Incident first responder represents the responsibility of CSP to make an immediate response before data vanishing which affects performing the data collection. The Role management challenge is the difficulty of recognizing the data owners and their real identity, which is not related to their authenticated credentials. The Last challenge is the training, as cloud forensics investigator requires special mandatory skills and training such as: IoT, microservices, virtual machine environment, mobile, network, and database forensics which require forming a team of experts.

Digital Forensics Research Workshop

Digital Forensic Research Workshop (DFRWS) is a conference that was held in 2001 and recurred every year to develop last updates concerning digital investigations. It proposes academics and practitioners’ approaches aligned together for achieving best practice methodology in cloud forensics. The investigation phases consists of six phases as follows:

In phase one, Pre-Process, the investigator creates an action plan for the investigation roadmap. It contains the incident scope, the services affected, the type of damage, and the available data for operation and evidence. Phase two, Identification, detects anomalous patterns and identifies the sources of valuable information. This phase distinguishes between useful and useless evidence. It chooses based on relation to incident types or attacks. Phase three, Acquisition & Preservation, concerns keeping the evidence’s availability and integrity in the live scenario. It preserves the crime scene with no changes as much as possible. It could be achieved by protecting valuable data before vanishing. For example, it keeps network traffic metadata in a meaningful form in a database, gathers data from microservices, extracts dump from VM, swaps files during the incident, and saves it in a safe location. Phase four, Analysis, is responsible for presenting primary data based on its type or source. It builds several pieces of the story about what and how digital crime was done. Phase five, Evaluation, involves a logical comparison of every piece of evidence that comes out of the analysis phase and determines its impact.
on the incident, identifies evidence pattern validation, and recovery of hidden/encrypted data. Phase six, the Presentation and post-processes, aims to close the investigation whether by presenting the evidence in a CoC form or dropping the case due to insufficient evidence.

2.2 Literature Review

Insight of DFRWS phases, researchers published many papers. They succeed in resolving six out of nine challenges from the NIST report. We classified the literature results into four deliverables. First, a central server is used for collecting artifacts in one location. Second, provide forensics as a service rather than handling incidents individually. Third, enhance analysis using machine learning and AI techniques. Fourth, use encryption methods to keep data integrity. These deliverables resolved the following challenges: architect, anti-forensics, role management, first response, training, and analysis. Data collection, standards, and legality are yet to be resolved. Table 1 depicts a comparison between several frameworks regarding the solved challenges and their subcategories. In addition to NIST’s subcategories, we invent some subcategories that, from our point of view, maintain the CoC to have trusted evidence to be presented at court such as: for data collection (full data evidence), for Standardization (API G.W, network metadata, PaaS information, SaaS MS, IaaS VM), for Legality (chain of custody), for Anti-Forensics (encryption and decryption), for first Response (postmortem, live, security), and finally, for Analysis (normalization, and machine learning).

Table 1 Literature Frameworks Comparison

<table>
<thead>
<tr>
<th>Subcategories</th>
<th>Data Collection</th>
<th>Standardization</th>
<th>Legal</th>
<th>Architecture</th>
<th>Anti-Forensics</th>
<th>Role Mgmt</th>
<th>First Response</th>
<th>Train</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Data Integrity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[6]</td>
<td>Data Available</td>
<td>Full Required</td>
<td>NT Metadata</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[13]</td>
<td></td>
<td></td>
<td>PaaS Info</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[15]</td>
<td></td>
<td></td>
<td>SaaS MS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[16]</td>
<td></td>
<td></td>
<td>IaaS VM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[17]</td>
<td></td>
<td></td>
<td>Law Privacy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[18]</td>
<td></td>
<td></td>
<td>Law Enforcement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[19]</td>
<td></td>
<td></td>
<td>Data Privacy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[20]</td>
<td></td>
<td></td>
<td>Data Privacy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[21]</td>
<td></td>
<td></td>
<td>Chain of Custody</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[22]</td>
<td></td>
<td></td>
<td>Data Segregation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[23]</td>
<td></td>
<td></td>
<td>Data Provenance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[24]</td>
<td></td>
<td></td>
<td>Encrypt Evidence</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[25]</td>
<td></td>
<td></td>
<td>Decrypt Evidence</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[29]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For using a central server, many solutions were presented by researchers [15, 16, 17, and 18]. It resolved the architect and role management challenges. In [19], the authors proposed a schema called “Secure...
Log”. It uses the central server concept but is still limited only to the systems that include logs. The proposed solution in [6] segregates the collected data before storing it in a unified location for later usage by all forensics activities. It also resolves the role management challenge, as it assigns the forensics mission to the CSP instead of an outsource investigator. Another solution proposed in [20] provides an algorithm to overcome anti-forensics attacks. It secured data transfer using Modified Elliptic Curve Cryptography (MECC) and Deep Learning Modified Neural Network (DLMNN) classifier. The algorithm processed the transferred data and identified the compromised ones. Consequently, the training challenge was resolved through the involvement of a highly qualified security team by the CSP to manage the forensics server. Finally, the researchers in [21] suggested placing the forensics server on the ISP side regardless of the limitation in accessing resources.

Due to the CSP’s responsibility and the availability of the evidence in a central server, the first response challenge is handled. It directly involves the first reaction [13] when the incident occurs or at the time when the incident is reported. Obviously, the first response is usually triggered to secure systems once the threat is detected in a live scenario. It also reacts in a postmortem scenario by providing the investigator with valuable information. In [22], the Event Reconstruction based on timeline methodology is discussed. It represents the backbone for building meaningful evidence, not only for forensics but also for providing a first response. Finally, in [23], the authors exposed the VM’s snapshots that could be taken and stored in a safe place before incidents occurred. This involves a huge effort and costs, further, the solution needs unlimited storage to cope with VMs versions.

Researchers work for data integrity by implementing Blockchain [11] to maintain the CoC for many devices/systems. The authors implemented Blockchain as a solution called Internet-of-Forensics (IoF) [24]. The IoT serves in a fog network with a heterogeneous nature equal to the microservices environment. They keep the records chain, starting from the data collection, provisioning, analysis, and evaluation until the presentation at court. However, one block of the evidence chain might not be enough in court cases. For example, when a VM incident occurs and logs are preserved, the logs are useless unless it contains other evidence like firewall traffic and the registered user’s information who owns the VM. In addition, the registered user’s information must be proven to be related to human identity not to unknown sources or internal CSP traffic. This relation could be proven by official national identification or payment card registration. Hence, in literature, they used the Blockchain to keep the integrity of a single block of evidence which does not build a comprehensive crime scene. This results in ignoring the evidence by the court.

New technologies aid in facilitating and developing efficient analysis results. As an example, [25] used Deep Learning (DL) to enhance the forensics prediction on the Internet of Things (IoT) environment. Other researchers identify the normal and malicious data using the Machine Learning (ML) fusion methodology [17]. In addition, others used Artificial Intelligence (AI) to detect broken authentication [26]. Finally, all previous models of AI are supported with big data handling techniques for providing efficient results [27]. Whereas, dealing with all data types is essential for enhancing results.
As a part of resolving the Standardization challenge, the researchers collected data from few (but not all) related data sources which resulted in a broken CoC. Examples of data sources include: API G.W, network metadata, PaaS, and IaaS extracting snapshot dumps as evidence. For example, in [8], they collect their data from network metadata and VM snapshots. In [28], the authors used Variational AutoEncoders (VAEs) technique for sensing CPU and RAM usage to detect CPU-Miner and HTTP-flood attacks. As illustrated in Table 1, no one is exposed to the microservices data at the operational level which represents a gap in the Standardization challenge. In our work, we will use microservices data in addition to the other data sources.

The legality challenge is not solved yet. All efforts are useless unless the court accepts the CoC and applies its authority on the trial. Hence, [21] discussed the law enforcement by authorities in digital crime cases. The solution is based on the collaboration between technicality, organizational, and legality aspects. They presented four scenarios to enforce law implementation. First, determine the physical location of the data center that hosts the digital evidence. Second, determine the physical location of the CSP headquarters. Third, determine the location of the end-user where the crime has a final effect. Finally, determine the nationality of the criminal or the victim, regardless of the location of the crime. There are three essential legal challenges facing cloud forensics: loss of location (distributed artifacts), data ownership (accessibility and credentials), and confiscation procedure (data acquisition). Insight into the USA and European legal frameworks, they discussed law enforcement by authorities [21]. For instance, in 2018, US federal law enforced “Microsoft Ireland” to provide data stored in servers located in the US or foreign countries [29]. The collected data contains ownership credentials, and there were legal procedures for acquisition. Finally, they presented four scenarios to enforce law implementation according to location and nationality: the physical location of the datacenter that hosts the digital evidence, the physical location of CSP headquarters, the location of the end-user where the crime has a final effect, and the nationality of the criminal or the victim, regardless of the location of the crime. This shows that effective cloud forensics needs international collaboration and third-party interference.

Solutions proposed in the literature focused on covering several challenges. Hence, due to the lack of resolving all challenges raised by NIST, there was no complete solution. An incomplete solution means that the artifact results have a broken chain of custody resulting in refusing to use these artifacts in the court as valid evidence. Many reasons led to this situation. First, the provided data is not enough to know how the full incident took place. Second, the provided evidence does not have a proven protection method, whereas there is no validation method to protect artifacts from being forged. Finally, the output report is not accepted by the court due to insufficient legality requirements. In the next section, a detailed description of the proposed framework is given.

3. Microservices Forensics As A Service (Msfaas) Framework

Conventional digital forensics methodologies became unreliable. Hence, we designed the MsFaas framework to be provided by the CSP. It aims to facilitate the mission of cloud forensics investigators and to provide a validation tool for microservices. The framework collects all meaningful data from multiple
sources such as registered user information, network traffic, firewall logs, Machine Learning (ML) output values, API gateway’s streamed data, and the microservices hashed data sent for validation. MsFaaS investigation is inspired by DFRWS phases and uses the event reconstruction stages. The proposed solution provides forensics evidence for live and postmortem scenarios. It protects microservices from MITM attacks. MsFaaS produces three services: end-user SLA agreement for subscribed customers, a forensics report certified by the local authority where the victim is located, and internal CSP auditing for users and systems activities. The main goal of our proposed solution is to maintain the CoC to be able to create certified forensics reports that support the final decision in a trial. The report is secured by blockchain technology. We focus on including all data types to complete a comprehensive crime scene. Since, the solution is a collaboration between technicality, organizational, and legality aspects, therefore, we designed MsFaaS as a technical solution for the first and second legal scenarios presented by [29] as previously mentioned. We target the case of law enforcement when the evidence server or the headquarter of the CSP is in the same legal domain of the court. Our methodology is to collect all previous ideas in one framework and resolve uncovered challenges such as: data collection, standardization, and legality. For instance, we used the idea of a central server for collecting evidence as presented in [15, 16, 17, 18]. We provide CSP forensics as a service; it is inspired by a client service named LAUXUS proposed in [6]. In addition, we used the interaction from a third-party to achieve evidence credibility as proposed in a system named TamForen [13]. The framework is to be implanted by the CSP’s cyber security team, and it applies specific organizational procedures incorporating a third-party governmental reference to the USA case presented in [29]. It uses the investigation phases presented by DFRWS and is supported by Event Reconstruction methodology as shown in Fig. 2.

Technically, the phases shown in Fig. 2 go as follows: Phase 1 sets all data sources related to the incident based on all up-to-date cloud threats scenarios. It plans for proper responses for each type including zero-day attacks. In phase one, the process of event reconstruction starts with step A (data collection). Accordingly, Phase 2 identifies, filters, and segregates different data types based on their sources and types. Also, based on ML methodologies, it performs the first response as a security countermeasure or proceeds with event construction- step B (data segregation). Phase 3 is the core phase where we preserve the data with keeping its CoC. It converts the collected metadata from raw form into a human-friendly form using the event construction step C (Extra Attribute), and step D (Normalization), then, step E which keeps data integrity using Blockchain and replicates the data to a governmental third-party location. Phase 4 generates analysis reports using deep-learning methodologies that provide clear pieces of relevant evidence. Phase 5 examines the logic between all related incident pieces and evaluates the comprehensive image of the crime. Finally, Phase 6 generates the forensics report which could be certified by validating the report’s hashed values to its consort at the replicated third-party server. Consequently, the certified report applies the low requirements.

### 3.1 MsFaaS Framework

The solution aims to facilitate the mission of DFI as well as the tasks of the internal security team. It provides evidence for live and postmortem attack scenarios. In addition, it protects microservices from
MITM attacks by providing a validation tool for microservice operated by the customers. To achieve these targets, MsFaaS provides three services: end-user SLA agreement for subscribed customers, a forensics report certified by the local authority where the victim (customer) or the CSP's datacenter is located, and internal CSP auditing for users and systems activities. The following are the three services in detail.

The first service, the End-User SLA, agreement, provides the customer with two facilities. The first facility is the validation method for his MS operation. For example, MS sends hashed values for “Order” to keep data privacy. Then, the process will proceed only as per MsFaaS validation. The second facility is to monitor all activities related to his workspace, then, produces a forensics report for all backend hidden activities. This report is certified by a local authority where the CSP’s datacenter is located.

The second service, Forensics Report, which is created by MsFaaS, gets a verification certificate by comparing the report records hashes to those previously replicated to the third-party. Hence, the third-party can verify the presented report without having any clear text in its database. For example, NTRA in Egypt should keep the required data for forensics purposes as per the “anti-digital data crimes” law.

The third service, Internal Auditing, involves using a massive amount of data transactions inside the CSP’s local area network. This direction helps in controlling internal threats against CSP assets and protecting clients’ data as per GDPR.

Following are the detailed steps for: the needed pre-processes, the event reconstructions, and the presentation of trustful evidence at the trial.

**Pre-Process Phase**

A set of prerequisites is defined by the CSP. It is essential to reconstruct the events in a way that satisfies the requirements for resolving the data collection, standardization, and legality challenges such as: data retention policy, type of data sources, data preserved format after normalization, keeping data integrity, keeping customers’ data privacy in CSP, and security of transferred data to the third-party. These technical and organizational prerequisites support the legality of multi-jurisdictional cases and multi-tenant environments [30] as follows.

For Legality purposes, based on the framework’s vision a set of rules serves the organizational and technical aspects. A retention policy keeps artifacts’ history for satisfying the organizational part. The policy is powered by the client’s SLA contracts with local authorities under international law [29], and System Operational Procedures (SOP). In addition, the technical part is fulfilled by implementing a local forensics server as an evidence store; and a third-party server at the local authorities to validate the evidence's integrity against forgery. The SOPs are secured using a secure site-to-site VPN tunnel. The tunnel transfers the replicated hashed data between the CSP local environment and the third-party storage.

The purpose of the Standardization process is to convert the data collected from different cloud platforms into a unified standard form. The process includes metadata identification, segregation,
normalization, and examination before analyzing presented evidence. An example is the conversion of the artifacts collected from SaaS, PaaS, and IaaS platforms into meaningful pieces of data.

The Data Collection process aims to identify the required artifacts. As a part of Event Reconstruction, it fulfills the “Step A-Data Collection” (Fig. 2). This is in the case of live incident investigation where the crime scene was not modified. In contrast, postmortem scenarios involve evidence vanishing. The targeted VM or container could be deleted, or the logs could be modified. Therefore, to implement an efficient data retention policy, we collected data based on the list of cloud attack scenarios mentioned in [23]. The required data types to be collected are as follows:

- Sniffed row metadata network traffic including all internal and external data packets.
- Firewall logs, users’ activities, and unknown sessions.
- Cloud service user’s identity involves IP address, hardware MAC Address, credentials, used protocol, used ports, location, name, and payment information (ignoring credit card number, secrets, and password).
- Events stream generated by the API’s getaway.
- Virtual Machines metadata traffic and dump images (swap files vRAM, vHD) in case of incident handling.
- Microservices subscribed to the MsFaaS “Validation Service”, storing hashed data for later validation.
- Hardware performance statistics, CPU, RAM, containers HTTP flow at the API’s interfaces.
- Output values and patterns generated by previous machine learning prediction.

By collecting all recognized data types from all available sources. We are extremely narrowing the gap of the data collection challenge and creating strong data correlation. Hence, every piece of evidence from SaaS, PaaS, and IaaS and all data sources build a full picture of a crime scene.

Identification Phase

This phase mainly aims to distinguish between valuable and useless data types. As a part of Event Reconstruction, it fulfills the “Step B-Data Segregation” (Fig. 2). Hence, the phase segregates and classifies the collected data based on its source and the correlation to each other. Big data streams sniffed from network nodes by Intrusion Detection System (IDS) or events extracted from API’s gateways need to be handled and transformed from row metadata into separated records. For instance, all data related to one tenant are distinguished by the instance unique Id Instance_uuid [6] and associated with the customer’s identity. Inside one tenant, different microservices act individually and could be recognized as a threat. We correlate the existing VMs and containers to: UUID, Firewall logs, and user information before it might be deleted or moved to another node in a different geographical location. This is done using anomaly-based or supervised machine-learning classification algorithms to classify massive metadata streams in clear records under specific categories. Finally, each bunch on recognized logs does
not build the full image that tells the full story, but it is useful for future normalization processes and help in analysis.

Preservation Phase

This phase is the most important phase since it the full image is constructed while keeping its integrity. We create the documentation for the full processes that build a tight CoC for supporting the issuance of a certified report. This phase consists of event reconstruction steps and uses three dedicated storages. The steps are: C-Extra Attributes, D-Data Normalization, and E-Data Integrity (Fig. 2). Data resides in three databases as follows: a relational database at the CSP, a non-relational database that contains a hashed copy of every piece of information at the CSP, and the database at the third-party that is governmentally authorized to certify the hashed values. The process starts in the following sequence.

Step C-Extra Attributes:

Based on the threat type, the records are collected into pieces of evidence. For example, DDOS attacks require adding multiple sources of connection whereas several IP addresses share in pushing a bunch of HTTP packets flow. Consequently, the hardware performance of the CPU, RAM, and API interface reaches their threshold which results in a service outage. On this occasion, it is obviously useful to add extra attributes like the direction of data flow, the timestamp, authenticated user login credentials, payment identity (no sensitive data), the used Token, and the used protocols. These extra attributes aid in identifying whether the attack is from inside or an unknown external source, whether it is a malware act from inside the VM due to misconfiguration (lack of antivirus) or a hacking action. Despite some of these attributes are not mandatory, this process outputs enormous pieces of evidence for the normalization step.

Step D-Data Normalization:

The main aim of this step is to attach all evidence pieces to a timeline as in [22]. This is done to transform the row metadata from separated rows into a friendly human form. The data is stored and categorized in a relational database presenting understandable behavior. Hence, it is rated based on risks and severity. For instance, the security category contains events related to login and logoff, while categories related to the user behavior contain: used file names, protocols, targeted IPs, or websites they already categorized as proxy, hacking, governmental, and education categories. Therefore, it will be easy to understand the risk of each behavior and categorize it. Therefore, a collection of artifacts is drawn on a timeline based on Greenwich Mean Time GMT) to form meaningful records as <Time-Description>. The reason for using GMT time is to mitigate the risk of performing actions from many destinations and using the differences in time worldwide. The data collected from multiple data sources is stored in two storage formats as illustrated in Fig. 3. First, the normalized data is stored in the relational database inside the CSP for internal monitoring and machine-learning activities. Second, the hashed version is stored in a non-relational database that has been replicated to the third-party. Consequently, the use of
hashed values has many advantages such as reducing the storage volume, keeping data privacy, and allowing the third party to issue validation certificates based on the hashed values.

**Step E-Data Integrity:**

We maintain data integrity in two directions. First, Blockchain technology is used on the data located in a non-relational database; in addition, day-to-day hashed values are replicated to a database located at a local third-party. The blockchain is to ensure the immutability of the hashes against falsifying. Hashes are stored in blocks; the block header contains: the hash value of the previous block, a time stamp, the Merkle hash root which is shown in Fig. 4, a nonce, and other information. Thus, a chain of hash blocks is created forming the Blockchain; any insignificant change in any block will be easily discovered due to the use of the previous block's hash. The second direction involves using an IPsec site-to-site VPN tunnel to securely replicate hashed values to a third-party organization. The targeted third-party is a local governmental organization that will be authorized to validate replicated data upon request.

Another integrity method is provided as a service by MsFaaS, where the customer sends a hashed copy of any data that needs to be validated at a later stage. For Example, in the case of microservices in an eCommerce application, it provides a validation method. The hash values are stored in a database under certain categories (for example Purchase Order) which later will be validated in the delivery or transfer out of the store operation (explained in detail in the use-case sub-section).

The integrity of one piece of information does not guarantee the coherence of the crime story which ends up in a broken CoC and rejection at the trial. Therefore, in MsFaaS, we replicate the full related data (SaaS, PaaS, and IaaS) which will be available for validation and certification by the governmental authority upon request. This theory is supported by two examples that will be discussed in the presentation phase: as per Egyptian law dealing with the National Telecom Regulation Authority (NTRA), and as per the US federation law.

**Analysis Phase**

The analysis phase is the first interface to the DFI work where he/she extracts pieces of evidence. It is an essential workspace for understanding what, when, and how the incident occurs; and the base to provide accurate outputs to the presentation phase. In MsFaaS, the analysis is a starting point for decision-making and incident response. We used a massive bunch of data from different sources to decide the required action in this stage. The variety of data types forced us to use modern technologies such as supervised machine learning algorithms that produce accurate predictions and other methodologies from current literature. Our contribution is an anomaly-based detection system consisting of four steps to differentiate between benign traffic patterns and suspicious behavior. The process steps are as follows: dataset preparation, correlation feature engineering, model training preparation, and classification. The next step is decided based on the output of the previous processes. In our work, we used a dataset named CSE-CIC-IDS-2018 [31] which is provided by the Canadian Institute for Cybersecurity. It was created by capturing all network traffic during ten days of operation inside a controlled network environment on
Amazon Web Service (AWS). The data is captured in row format and placed into excel files. It contains 80 features (data types).

In the first step, dataset preparation represents pieces of information such as timestamps, protocols, ports, flow duration, and packet size. We used Python programming language to import the data file from a Microsoft Excel sheet. The used files include many records that show the benign traffic as well as DOS, DDOS, brute force, Bot and web attack patterns. Then, we prepare the data by removing unnecessary features. Figure 5 shows a sample of patterns that differentiate between benign traffic and the abovementioned attacks.

In the second process, correlation feature engineering, a relation comparison is done using Python programming language “Spearman rank correlation” as shown in Fig. 6. It results in plotting a heatmap diagram with the relationship among all pieces of information as illustrated in Fig. 7, and in a dendogram diagram which is depicted in Fig. 8.

In the third process, model training preparation, we filtered the most relevant 30 features based on the lowest variance as shown in Fig. 9. The resulting output has the 30 most correlated features and forms specific patterns.

The fourth process is the classification step. Each classified pattern identifies a certain attack type. Then, the decision takes place based on a fact factor, whether the traffic pattern is malicious or benign. For example, attacks such as: Brute Force, DoS, DDoS, Heartbleed, Web, Infiltration, and Botnets trigger an incident response.

On the other hand, normal traffic is recorded with other data from different resources for future forensic incidents.

Another example of detection is included in the model such as an Event Sequence Graph (ESG) which captures the behavior of a variety of interactive systems as in [32]. Variational AutoEncoders (VAEs) technique is used to sense CPU and RAM usage to detect CPU-Miner and HTTP-flood attacks.

**Evaluation Phase**

Maintaining CoC and ensuring the logical sequence of stories are the main DFI tasks in this phase. The investigator validates the logical coherence of the data coming out from the analysis phase and makes sure that all collected evidence has a legal collection procedure. The correct sequence for the logs on a timeline shows a logical events scenario. For example, some events logging into the cloud with an account need to be followed with an action using the same credentials. In the case of data encryption, DFI can use Forensics Tool Kit (FTK) [33] to decrypt an encrypted piece of evidence. The investigator also needs to associate every piece of evidence with its legality procedures under SLA, local authority contract, or international law. MsFaaS obviously facilitates this mission as it already provides scenarios supported with complete evidence from different data sources.
Presentation Phase

MsFaaS service aims to provide the investigator with a complete analysis report to be presented as evidence at the court. Hence, law enforcement takes place. As per [29], the US federal law CLOUD enforced “Microsoft Ireland” to provide data stored in multiple servers. Some are in the US, and others are in foreign datacenters in different countries. The authors identified four legal occasions to enforce the law when it takes place in the same location of the court: the physical location of the datacenter that hosts the digital evidence, the physical location of CSP headquarters, the direct consequence place or final effects of the crime, location of the end-user, and the nationality of the perpetrator or the victim, regardless of the crime’s location. In parallel, as per national law in Egypt, organizations are responsible for taking necessary action to help against digital crimes. Thus, the report needs governmental validation to be considered as evidence at the trial. Since the report data is validated using the hashed copy, the report gets certified and guaranteed that the analysis report is not forged. In the worst-case scenario, the report will be used as an extra clue in the trial. Hence, the framework is completed as shown in Fig. 10. Whereas the legal challenge is resolved in terms of privacy and CoC, the investigator can provide a certified report to the court.

3.2 Use Case (eCommerce)

In this subsection, we present the usage of the End-User SLA. The service provides a validation method for any workflow that needs validation. The service validates critical functions during a workflow process. In this case, we simulate eCommerce workflow and focus on the process of payment and delivery as microservices in a SaaS application. The service makes sure that the data (Order) that was transferred between the order microservice and the delivery microservice is not modified by any method. The (Order) JSON object [34] has the details of the sales order including the amount for payment and the items to be delivered. Hence, the SLA should ensure that the delivery items’ quantities are identical to what was paid in the payment microservice. The service ensures that it was not modified (hacked) by internal or external interference. In the case of occurrence of a hacking incident, the MsFaaS contains the required evidence to simulate the complete crime scene.

The system consists of three microservices (Containers- with APIs): Order-MS, Delivery-MS, and MsFaaS-MS. The APIs are considered as a backend as illustrated in Fig. 11, and they are coded with ASP.Net Core version 6 and SQL standard database. On the frontend, we used the “Postman” testing tool to simulate the client side and hacker activities.

There are two order types, normal (in green) for regular operation and a modified order (in red) which is affected by a hacker. The postman acts as a MITM hacker [35]; it simulates traffic interception and modifies the order object prior to forwarding it to the Delivery-MS for the process. The modification purpose was to increase the number of items to be delivered. In real life, interception could be done by Kali Linux (Bhargav and Akif) and the “suite prep” tool. The attacker spoofs the MS sender’s IP address and intercepts the sessions. Then, he/she sniffs the traffic and modifies the payload before being sent to the receiver for processing.
The test proceeded with the two scenarios. The first workflow, the normal process goes through the steps as follows:

Step 1: The postman sends an order to Order-MS which in turn converts the order object to a Hash.

Step 2: The hash value is forwarded to the MsFaaS.

Step 3: The Hash value is stored in the database under the orders category.

Step 4: The Order-MS forwarded the normal order (Jason object as shown in the green order in Fig. 11) to the Delivery-MS to process the delivery operation.

Step 5: the Delivery-MS verified the order hash with the MsFaaS prior to processing the delivery.

Step 6: After order validation, the order is delivered to the client successfully.

The second workflow, the MITM attack simulation uses the postman to modify the same order and sends it to the Delivery-MS in (step 4). The Delivery-MS sent the order hashed value to the MsFaaS for verification and received a failed response as the hashes are not identical due to the modification. Consequently, the delivery operation was interrupted, and an alert was sent.

The test succeeds in showing one case of many attack scenarios which could be implemented on the cloud and microservices environments. It shows the usability of valuable stored data (the hash value in this case) which could be correlated to enormous attack types or malicious behavior. The result is that many validation applications could use the same service for integrity purposes.

### 3.3 MsFaaS Impact

The implementation of the MsFaaS framework affects the investigation quality whereas it provides trustworthy evidence. The collected information from the microservice needs to be supported with all related data from all surrounding cloud environments. Hence, we created the below set of questions included in the forensics report:

- Who did the incident? Is he registered or unknown?
- How did he make it? Explain technical methods.
- When? Presents the logical incident coherence.
- What is the impact? Shows what the effect is on both the criminal and the victim as well.
- Can anyone else does it or forge it? explain the possibility of data forging and evidence integrity.
- Is the evidence collected lawfully? Exposed to the legal part of the full process of the system.
- What are the data collected to ensure providing the comprehensive image (Firewall traffic / User Credentials / Container UUID / MS hashed values/timeline)?
MsFaaS answers previous questions and supports the postmortem forensics scenario with a full set of information. It builds the crime scene and ensures its integrity which is the most challenging aspect of cloud forensics. Answering these questions provides efficient internal monitoring and control for activities inside CSP’s trusted environment and in the untrusted customer environment. This helps in creating a well-controlled and secured cloud environment. In addition, the proposed framework solves the following challenges: legality, standardization, and data collection. Consequently, MsFaaS is a base for implementing an immediate response service to prevent hacking or any other violation incident.

4. Conclusions And Future Work

Due to the difficulty of the cloud forensics service, achieving a complete forensics service is not claimed by literature yet. Therefore, we presented MsFaaS hypothetical framework to prove that the forensics-as-a-service is an applicable target. The solution has a tangible result whenever the cloud service provider CSP sponsors it and implements it as a service. As a prerequisite in our framework, CSP can apply a set of rules and regulations to cover all challenges raised by NIST. To present trusted evidence to the court, we kept the chain of custody by implementing the proposed framework MsFaaS. It presents valid technical solutions based on previous research and innovative ideas like collecting a wide range of data types; normalize them by correlating and classifying them with machine learning algorithms. The framework facilitates the digital investigators’ mission in the SaaS microservices environment, especially when it comes to investigating postmortem cases. In addition, it also provides a validation method to protect microservices from man-in-the-middle attacks. Hence, a complete framework tackles the challenges such as Cloud Architecture, Data Collection, Standards, Training, Legal, Anti-forensics, and Incident first responders. Finally, the proposed framework under CSP sponsorship answers a set of questions that were asked in the forensics report. Future work needs to go in two directions: law enforcement should obligate the CSPs to implement the roles and regulations using such innovative solutions. The second direction is to develop multiple response actions based on different threats and scenarios. The attacks such as Brute Force, DoS, DDoS, Heartbleed, Web Attacks, Infiltration, and Botnet are already recognized, and there are best responses. Hence, MsFaaS could be developed to be a comprehensive security service. It could be extremely powered by many technologies such as deep learning and artificial intelligence in the future.

Statements And Declaration

Funding Information

The authors have no relevant financial or non-financial interests to disclose.

Author’s Contributions

Abubakr: First writing of the manuscript, puts the main ideas of the manuscript, and applies any requested modifications.
Heba Aslan: Reviewed the research plan, participated in all results, and final revisions to submit.

Research Data Policy and Data Availability

We used open datasets such as the CSE-CIC-IDS-2018 on AWS [31].

Ethics

“Conflict of Interest: The authors declare that they have no conflict of interest.”

This article is original and contains unpublished material. The corresponding author confirms that all other authors have read and approved the manuscript and no ethical issues have been involved.

References


https://doi.org/10.1016/j.cose.2019.05.020

https://doi.org/10.1016/j.future.2021.02.016

https://doi.org/10.1016/j.future.2020.03.042


https://doi.org/10.3389/frai.2020.00004

https://doi.org/10.48550/arXiv.2104.13193


https://doi.org/10.1109/ICIRCA48905.2020.9182938.

https://drive.google.com/drive/folders/ 1HrTPh0YRSZ4T9DLa_c47IubheKUcPl0r

https://doi.org/10.1109/CALCON49167.2020.9106500.


35. Pingle, B., Mairaj, A., Javaid, A.: Real-world Man-In-The-Middle (MiTM) attack implementation using open source tools for instructional use. IEEE International Conference on Electro/Information
Figures

1. Provenance
2. Data Segregation
3. Multi-Tenancy

Figure 1

Mind Map Challenges Categories (NIST)

Figure 2

DFRWS Phases Supported by Event Reconstruction Steps
Figure 3

Collected Data Sources

Figure 4

Merkle Root

Figure 5
ML Process 1 – Dataset Preparation

```python
cluster_threshold = 1

corr = spearmanr(X).correlation
corr_linkage = hierarchy.ward(corr)

fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(25, 40))
sns.heatmap(corr, xticklabels=X.columns, yticklabels=X.columns, linewidths=.5, cmap=sns.diverging_palette(620, 10, as_cmap=True), ax=ax1)
dendro = hierarchy.dendrogram(corr_linkage, labels=X.columns, ax=ax2, leaf_rotation=90)
dendro_idx = np.arange(0, len(dendro['ivl']))
ax2.plot([0, 1000], [cluster_threshold, cluster_threshold], ':r')
plt.show()
```

Figure 6

ML Process 2 – Spearman Rank Correlation

![Spearman Rank Correlation Diagram]

Figure 7

ML Process 2 – Features Correlation - Heatmap

![Features Correlation Heatmap Diagram]
Figure 8

ML Process 2 – Classes Filtering Dendrogram

```
cluster_ids = hierarchy.fcluster(corr_linkage, cluster_threshold, criterion='distance')
cluster_id_to_feature_ids = defaultdict(list)

for idx, cluster_id in enumerate(cluster_ids):
    cluster_id_to_feature_ids[cluster_id].append(idx)

selected_features = [v[0] for v in cluster_id_to_feature_ids.values()]

selected_features = x.columns[selected_features].tolist()

print('selected features: ')
pd.Series(selected_features)
```

Figure 9

ML Process 4 – Classes Filtering Dendrogram
Figure 10

MsFaaS Framework

Frontend

1 - Submit Order

Client

Order

Delivery

2 - Order

Modified Invoice
Invoice# 12345
Amount: $2k
Items Modified: 4
Address: Cairo

MITM Modification

3 - Save Hashed Order to DB

6 - Delivered

Yes

Validste

Delivery MS

Alert MS

No

Normal Invoice
Invoice# 12345
Amount: $2k
Items: 2
Address: Cairo

4 - Send to Delivery MS

Order MS

Backend

Alert MS

Figure 11

POC for eCommerce Validation