Journal of Information Security and Applications 42 (2018) 18–28

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

Towards a practical cloud forensics logging framework

Ameer Pichan*, Mihai Lazarescu, Sie Teng Soh

Department of Computing, Curtin University, Kent Street, Bentley, Perth, WA 6102, Australia

ARTICLE INFO

Article history:

Keywords:
Cloud computing
Cloud forensics
Digital forensics
Digital evidence
Logging
Cloud service provider
Cloud service user

ABSTRACT

This paper exposes and explore the practical issues with the usability of log artefacts for digital forensics in cloud computing. Logs, providing detailed events of actions on a time scale have been a prime forensic artefact. However collection of logs for analysis, from a cloud computing environment is complex and challenging task, primarily due to the volatility, multi-tenancy, authenticity and physical storage locations of logs, which often results in jurisdictional challenges too. Diverse nature of logs, such as network logs, system logs, database logs and application logs produces additional complexity in the collection and analysis for investigative purposes. In addition there is no commonality in log architecture between cloud service providers, nor the log information fully meets the specific needs of forensic practitioners. In this paper we present a practical log architecture framework, analyse it from the perspective and business needs of forensic practitioners. We prove the framework on an ownCloud - a widely used open source platform. The log architecture has been assessed by validating it against the Association of Chief Police Officers Good Practice Guide for Computer-Based Electronic Evidence guidelines. Further validation has been done against the National Institute of Standards and Technology published report on Cloud Computing Forensic Challenges, i.e., NISTIR 8006. Our work helps the forensic examiners and law enforcement agencies in establishing confidence in log artefacts and easy interpretation of logs by presenting it in a user friendly way. Our work also helps the investigators to build a collective chain of evidence as well as the Cloud Service Providers to provision forensics enabled logging.

1. Introduction

Cloud computing has revolutionized the computing industry in the recent years. Cloud computing offers unlimited computing power and storage on a pay per service model, which enables the business to shift the Information Technology (IT) service expenditures from Capital Expenditure (CapEx) to Operational Expenditure (OpEx), resulting in rapid uptake of cloud computing services. The Bessemer Venture Partner (BVP) Cloud index (which tracks the top public companies), reported that the cloud market will triple in next five years reaching US$500 billion of total market capitalization by 2020 [1]. The BVP State of the Cloud Industry 2018 report predicts strong growth in cloud computing uptake and the rise of innovative cloud services (e.g., Serverless Computing, Payment as a Service) [2]. Also the Right Scale 2017 State of the Cloud Report mentioned that the cloud adoptions continues to grow and companies started running majority of their business applications in cloud computing environment [3]. Despite the fact that the cloud computing is uniquely susceptible to the confusion and hype that surrounds it.

Though the cloud computing offers significant benefits, there has been growing concern about the security, privacy, legal, and jurisdictional aspects of cloud environment and the way the cloud computing stores and process customers data [4]. Further to that researchers have pointed out that the cloud infrastructure is not matured to support digital forensic needs as well, and identified issues and challenges associated with conducting forensics in the cloud [5–7]. Researchers also have noted that, till date, there is no vendor which facilitates the forensic investigation in the cloud [8]. National Institute of Standards and Technology (NIST) identified 65 cloud forensic challenges, in its report titled *NIST Cloud Computing Forensic Science Challenges* i.e., NISTIR 8006 report [9].

One of the challenges identified in the NISTIR 8006 report is the *Criminals access to low cost computing power* [9]. The availability of massive computing power at low cost is a motivational factor for malicious actors to use the cloud computing infrastructure to conduct cyber crime, or to store counter band materials [10]. It has been reported that Amazon cloud infrastructure has been used to store nasty SpyEye banking trojan and launch attack on financial institutions, affecting seriously the financial institutions in US, UK, Canada, Germany and Australia [11]. Criminals quickly disappear by

* Corresponding author.
E-mail address: ameer.pichan@postgrad.curtin.edu.au (A. Pichan).
URL: http://www.curtin.edu.au (A. Pichan)

https://doi.org/10.1016/j.jisa.2018.07.008
2214-2126/

terminating their account, but forensic examiners still should be able to trace the malicious act to the actors. The proposed model in this work helps to address the traceability of malicious activity, even after the actors have terminated their cloud services.

Many researchers have established cloud logging is an essential need for cloud forensics. Event logs, application logs, system logs, network logs are fundamental forensic log artefacts. By collating all the logs and putting them over a time scale helps to connect the chain of events. Therefore, many research work has been carried out in the related area; such as securing the logs, ensuring the integrity and trust worthiness of logs, secure transportation of log and enabling cloud to provide secure logging as a service [10,12–16]. However, none of the work looked from the angle of what forensic practitioners wants in the log. Therefore in this work, we examine the logging requirements for forensic needs which the law enforcement wants such that it can provide better value and practical benefit to the investigator. We propose a forensic enabled cloud forensic logging framework, and support it with experimental results. We then validate the framework using sample case studies and using Association of Chief Police Officers Good Practice Guide for Computer-Based Electronic Evidence (hereafter referred as ACPO guidelines) and NISTIR 8006. We further elaborate the model's capabilities, helping to build a corroborated chain of evidence in support of cloud forensics.

The rest of the paper is organized as follows: Section 2 provides the basic concepts and definitions. Section 3 describes related work conducted in this field and the motivation to carry out this specific research. Section 4 explains the methodology, architecture and environment used to carry out the experiments and to conduct the proof of validation. We describe the results and analyse of the findings in Section 5. Finally we conclude this paper in Section 6 along with the possibility of future work and Section 7 (Appendix A) lists the detailed test results.

2. Basic concepts

Cloud Computing. Cloud computing realizes the computing resources and IT services as a utility. Cloud computing benefits the Cloud Service Users (CSUs) by providing uninterrupted services with reduced or no maintenance overhead. It is a new way of service delivery model for computing resources and enables universal access, any time from anywhere over the Internet [17]. Amazon Web Services (AWS), Microsoft Azure and Google's App Engine are examples of Cloud computing.

Digital Forensics. Digital Forensics is an investigative process to determine and relate the evidence to establish factual information for judicial review. NIST defined digital investigation as *"the application of science to the identification, collection, examination, and analysis of data while preserving the integrity of the information and maintaining a strict chain of custody for the data"* [18].

Cloud Forensics. Cloud Forensics is the science and art of applying digital forensics in a cloud computing environment. NIST defines cloud forensics as *"the application of scientific principles, technological practices and derived and proven methods to reconstruct past cloud computing events through identification, collection, preservation, examination, interpretation and reporting of digital evidence"* [9]. The nature of Cloud computing architecture poses numerous challenges to cloud forensics in comparison to traditional digital forensics in an in-house IT systems [4,5,19].

Logging for Forensics. In a digital world, logs provides a systematic representation of the state of an object and the actions that has been taken producing a change in status of the object, generating events. Systematic and secure logging of those events, storing and making it available to the investigators, are an important and fundamental part of the cloud forensics. Because of the black box nature of the cloud, co-mingling of data, volatility, data integrity, jurisdiction, privacy of co-tenants etc. causes big challenges in producing a reliable logs for forensic purposes. Investigators have to depend upon the Cloud Service Provider (CSP) for the logs. Unfortunately there is no established process to verify that the CSPs are providing correct logs to the investigators either [10]. Despite, the logs provide highly valuable information to the investigators.

3. Related work and motivation

3.1. Related work

Forensic identification and data collection is a post crime activity, whether it is traditional forensics or cyber crime. In a cloud computing scenario, the evidence identification and collections is even more challenging due to ephemeral nature of cloud computing environment and geographic distribution of the physical systems [5,19]. The importance of cloud computing applications to provide some form of audit trail, as a digitally admissible evidence is of critical importance for cloud forensics [6]. However, researchers mentioned that cloud forensics is still in its infancy, neither the cloud providers nor the forensic community have yet put forward how they will implement the cloud platform forensic ready [19]. Therefore several researchers have looked at the problem of capturing trustworthy logs from multiple dimensions. Marty [14] provided a guideline for cloud application logging. Sang [16] described a log based approach for cloud forensics primarily for Software-as-a-Service (SaaS) and Platform-as-a-Service (PaaS) models, but the approach heavily depends on the CSPs providing support. In an attempt to find a solution for forensic investigation Zafarullah et al. [20] proposed a method for identifying and extracting log entries relevant to forensics from Linux operating system and security logs. They conducted experiments in Eucalyptus cloud environment and could produce fingerprints to reconstruct an event. Dykstra et al. [21] evaluated popular forensic data acquisition tools and proved that they can successfully return volatile and non-volatile data from the cloud, and examined various levels of trust required in the cloud. Further to that they developed an open stack tool, namely Forensic Open Stack Tools (FROST) to collect logs from virtual disks, application logs and firewall logs which works at the cloud management plane requiring no trust of the guest machine [22]. Zawoad et al. [23] proposed a Secure-Logging-as-a-service which stores entire virtual machines' logs and provides access to forensic purpose securely. They further expanded their work in which they presented a scheme for tamper proof secure logging, and proved that the integrity of the log can be ensured, even if the cloud actors such as CSP, the CSU and the investigator collude. The scheme ensures that any violation of the integrity property, can be detected during the verification process [10]. A layered cloud logging architecture was presented in the work of Patrascu et al. [15], including the way of monitoring activities in a cloud infrastructure.

Even before the advent of cloud computing, many research work have been carried out on the topic of secure logging for forensic purpose [13,24]. The secure logging principles outlined in those papers are valid for cloud computing environment too, though conducting various digital forensic process, especially the evidence acquisition is a complex process in cloud computing environment. A generic scheme that allows keeping the audit logs on an untrusted machine was presented by Shneirer et al. [24]. They proved that even if an attacker takes control of the untrusted machine the scheme ensures that the attacker gains little or no information from the logs and limits the attacker's ability to corrupt the log files undetectably. These principles can be extended for cloud

computing environment as well. It is to be noted that in digital forensics the time line of events is very critical. Any malicious attempt to alter the event time line seriously undermine the credibility of the evidence. To ensure the integrity and resilience of the time line of events, Battistoni et al. [25] proposed a novel architecture, which can very well be applied to cloud logging architecture as well.

Keeping the entire virtual machines logs even after client has terminated their CSU account could be costly for the CSPs. Legally valid forensic artefacts are not left around in a cloud environment once the perpetrator terminates his/her CSU account post committing the crime. For example one can create an Amazon free tier account and build an IaaS platform which can be used to launch cyber attack. The perpetrator(s) can easily terminate their CSU account and disappear into ether. It will be very difficult, if not impossible to collect any evidence of their crime. Therefore, it is very important to have the logs stored in persistent storage and make available to the investigators even after the CSU account has been terminated. Therefore, there is a strong need to create the logs outside the control and knowledge of the CSU, as well as preserve the evidence even after the CSUs' account has been terminated and the cloud resources released.

3.2. Motivation

Investigating inappropriate or illegal activities using cloud services are especially difficult, because of logging of data from multiple customers may be co-located and can quite well be spread across ever changing hosts, and that could be physically located in different jurisdictional area. It is critical to have an over arching service level agreement specifying contractual commitment to support digital investigation, along with proof of evidence that the vendor has been providing similar services. If not, it is safe to assume that digital forensic investigation may not be possible [4,12]. Further, few of the elements that makes the forensic harder in cloud computing includes, lack of standard interfaces and collaboration among providers, evidence segregation, lack of physical access and data recovery [26].

Though fingerprints of activity may be available in the host Operating System (OS), it is not easily accessible, as well those logs would be a collection of all significant activities happening in the virtual machines that the host OS is in control. Which also includes all the CSUs actions sharing the same OS platform, producing evidence segregation issues. What is logged in the host OS logs remains a proprietary info of the CSPs and varies according to the cloud architecture too. Often cloud providers implements propriety technology (e.g., Amazon Web Services (AWS) implements completely a proprietary Virtual File Management system), thereby making investigation task much harder even if disk image has been collected. Moreover, to access and collect those logs, CSP's co-operation and willingness is essential and in general is hard to come-by [8].

CSPs provide various types and levels of logging, but mostly for security and compliance reasons. Examples are AWS Cloud Trail [27], Azure Activity Log [28]. These logging mechanism provides a comprehensive logging describing 'who, what and when' performed an operation. But all such services are paid services and is controlled by the client. The customers have full control to enable and configure the logging services. Even though the logging provides CSU and system level activities, and is a vital tool to find out the activities happened in a customer's cloud space, nevertheless such tools are less sufficient from a forensic perspective. Mainly because the criminals are not going to configure and enable the logs, rather they would be doing the opposite and try to erase all the traces and evidence. Thereby substantiating the necessity of easily retrievable CSU specific logs, which are outside the accessibility of every CSU, stored securely in the CSP's environment. Same time ensuring that one CSU log do not co-mingle with other CSU's logs in a multi-tenancy environment and thereby safeguarding the privacy of co-tenants. Therefore, the preferred approach is to have a CSU action traceable logs, satisfying forensic and incident handling needs which can be easily acquired.

NISTIR 8006 mentions that *"an important source of forensic analysis is logs many of which may be available in cloud environment but may be hard to access or aggregate due to the segregation of duties among actors and lack of transparency of log data"* [9]. The report also talks about deletion in the cloud i.e., attributing the deleted data to a specific CSU is a big challenge. Therefore the logging for forensic purpose should include full CSU credentials and done without any CSU knowledge and beyond their accessibility, same time generating logs satisfying the legal needs. The Principle 3 of the ACPO states that *"An audit trail or other record of all the process applied to the digital evidence should be created and preserved. An independent third party should be able to examine those process and achieve the same result"* [29]. The principle emphasis the necessity of generating and preserving the evidence. Further Principle 3 states that the logs should be straight forward and easily described and can be understood by the Layperson [30]. Therefore, in simple terms the requirements for forensic logging can be summarized as follows:

- The ability to recreate the events by an independent third party.
- The ability to track events by connecting the dots in the chain of actions.
- The ability to attribute the action to a person.
- The non-interference of one CSU's activity with others - i.e., clear separation of every CSU actions.
- The ability to collect the evidence logs, soon the crime is detected, or when needed, preferably without affecting the cloud services or other cloud customers.
- The ability to go beyond jurisdictional boundaries.
- The ability to establish trust.
- The ability to ensure confidentiality and integrity of logs.

The proposed cloud forensic logging framework addresses some of the listed needs of the practitioners from a business perspective, and identifies the work done by researchers in support of other needs. In addition the framework presents a flexible, scalable and maintainable architecture. Therefore this work complements the work done so far in the same space.

4. Methodology

In the following sections we are proposing a practical approach to cloud forensics logging model. We then prove the model by implementing the model on an ownCloud instance. We validate the model by correlating the outputs of the model with respect to scenario based case studies and examining whether the model satisfies the criteria laid out in the ACPO guidelines and addresses the relevant challenges listed in the NISTIR 8006 [9].

4.1. The framework and architecture

The proposed cloud forensic log framework, termed as CFLOG consists of two main components, the CFLOG application and the log artefacts generated by the application.

4.1.1. The CFLOG framework

The main characteristics of the proposed framework can be listed as follows:

- The logging system has been designed to run outside the control and knowledge of the CSUs. Logs generated and stored out-

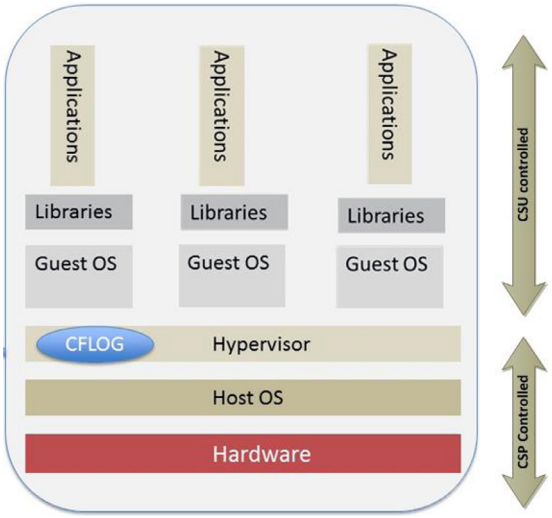

Fig. 1. CFLOG stack.

side the CSUs' preview increases the integrity and trustworthiness of the logs.

- The CFLOG application runs at the Hypervisor level on the virtual stack of computing environment, as depicted in Fig. 1. The figure represents a simplified version of cloud reference architecture [31]. In the architecture, the layers Hypervisor and below are controlled by the CSPs and not accessible to the CSUs, whereas the layers above Hypervisor are accessible to the CSU. We propose the CFLOG application to be embedded at the Hypervisor level.
- The architecture is very much scalable. Additional logging parameters and attributes can be added or removed easily. The architecture supports easy adaptation of specific forensic needs, compliance to standards or jurisdictional requirements.
- The architecture proposes to create one or multiple log files per CSU, thereby separating different every CSU actions from one another. The log files generated by CFLOG application are stored in pre-configured location, in the CSP controlled portion of the cloud stack. The method of separating each CSU actions, meets one of the key requirements of the cloud forensics and also makes it easier for the Law Enforcement Agencies (LEA) to collect and retrieve a particular CSU log. If all the CSUs actions were combined into one log file, that would create an enormous task of extracting the relevant parts of the log, removing the noise, to produce the data of interest. Modifying the log file contents can break the integrity. In any court of law, modifications of the logs will be often subjected to questionable doubts and can weaken the case significantly.

4.1.2. The CFLOG architecture

The architecture of the proposed CFLOG has been designed from a forensic practitioner business point of view. Specifically the forensic investigation seek to answer the six key questions or parameters of an incident *i.e., who, what, when, where, how and why* an incident took place [32]. The same six parameters also serves as the key elements to incident handling strategies [33], and identified that in the 'Assessment' phase of incident handling, the key task is to determine *who, what, where, when and how* an incident took place. The architecture laid out in this work seeks to answer these fundamental questions and therefore, supporting the incident response too.

The structure of the log file can be presented as a set of Log File Per User (*LFPU*), i.e., $LF = \{LFPU_x, LFPU_y, \cdots LFPU_n\}$, where $LFPU_i$ is the Log File corresponding to CSU *i*. Typically in an organization, there can be one root CSU or account owner and multiple CSUs under the root account using the same cloud platform. In such cases the Log File Per User, *LFPU* can be adapted as a set of $\{LFPU_{r,x}\} = \{LFPU_{1,1}, LFPU_{1,2}, \cdots LFPU_{1,n}\}$, where r is root CSU id and x is the id of CSU x. The LFPU name shall be created by using the combination of root id and CSU id and date time, which are then stored in a similar hierarchical folder structure. Thereby making easier to identify and locate the LFPU for a given CSU in the CFLOG stack.

Table 1
CFLOG entry parameter and description.

Parameter	Description
UTC_timestamp	Timestamp in UTC *(when)*
user	CSU id or CSU name *(who)*
source_ip	ip address of the source device *(from where)*
src_port	The port in which the source device is connected to
destination_ip	Destination ip address of the target host *(the target)*
local_time	Local time of the source device *(when)*
proto	Protocol used for communication *(how)*
file_param1	Contains an array of file parameters, *(data supporting CSU actions)* (optional parameters)
folder	Folder name of the file
file_name	File name *(objects or action parameters)*
size	File size
location	Geographic location
action	The action carried out *(what, how)*

The LFPU facilitates evidence segregation, in a multi-tenancy environment. Note that evidence segregation has been identified as one of the key challenges in cloud forensics [4]. The LFPU is written to a persistent storage, along with the file hash. LFPU is written in JSON format. JSON provides an easily understand structure for representing data. The generated logs i.e., *LF* are secured and can be transferred from CSP to the LEA using tamper proof secure protocol proposed by Zawoad et al. [10].

Each log entry *LE* contains a set of parameters. For user *i* LE_i consists of: $LE_i = [UTC\_timestamp, user, source\_ip, source\_port, destination\_ip, local\_time, proto, [[file\_param1, folder, file\_name, size], location \ldots\ldots], user\_action]$.

The parameters in LE_i can be added or deleted depending upon specific forensic needs, making our framework scalable. Table 1 lists some example of parameters and their meanings. Further, as shown in Fig. 2, each parameter is represented by a *key value pair*. The key-value pair of representing the data makes the file easy to parse and understand, as noted by previous researchers [14]. Moreover the key value pair helps the event reconstruction and incident handling.

4.2. System details and experimental environment

To build the cloud platform, we used ownCloud[1]- an open source product which enables building a cloud platform. ownCloud platform has been used by many researchers to run simulations and to validate their concepts and theory, including in the forensics space. To cite few examples, Martini and Choo [34] conducted the study on cloud storage forensics using ownCloud , Alex and Kishore [35] validated their cloud forensics framework using ownCloud. Rahman and Choo [36] used ownCloud platform for their work on Integrating digital forensic practices in cloud incident handling. Therefore, ownCloud is a proven platform for conducting such study.

We created a cloud environment using ownCloud (version 8.0), MySQL (version 5.6.17) and Appache web server (version 2.4.9),

[1] https://owncloud.org.

```
{“Records”:[
{
  "UTC_Timestamp": "yyy-mm-dd hh:mm:ss",
   “user”:  “User name”,
  “src_ip”: “xxx.xxx.xxx.xxx”,
  “src_port”: “nnnnn”,
  “destination_ip”:[
         “yyy.yyy.yyy.yyyy”,
      ],
 “local_time”: “yyyyy-mm-dd hh:mm:ss”,
 “proto”: “protocol”,
 “file_params1”: {
        “folder”: ”folder name”,
        “file_name”: “file-name”,
        “size”: “file size”,
       “location”: “geographic location”,
      },
 “file_params2”: { ……..},
 “action”: “user action”
},
{……….
  ……..},
{…….
   ……..},
 ]
}
```

Fig. 2. CFLOG structure.

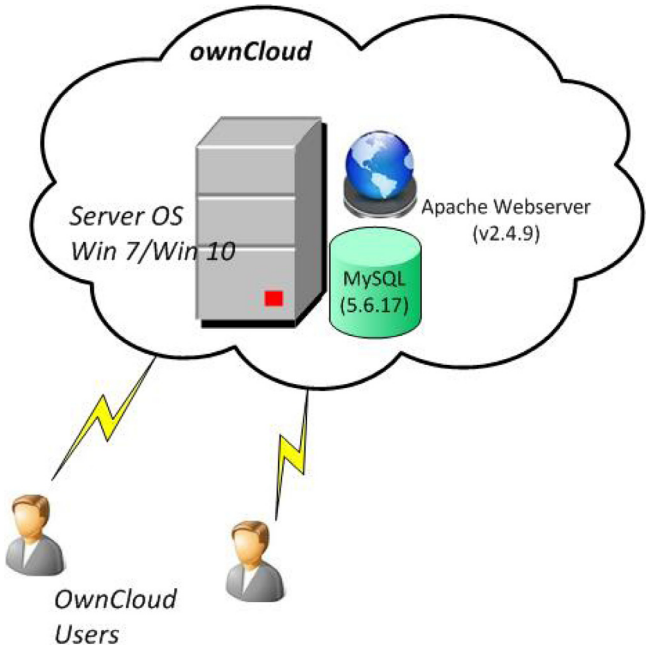

Fig. 3. Experimental environment.

with PHP (version 5.5.12) running on 64 bit WampServer stack. The virtual environment is running on a host computer, with Intel (R) Core I7, 3.40 GHz CPU and 8GB RAM, with 500GB hard drive, with 64 bit Windows 7 (Enterprise Edition) or Windows 10 as the host operating system. Fig. 3 illustrates the experimental set up. Cloud Forensic log application (CFLOG) is written using PHP and deployed as part of ownCloud platform. The CFLOG captures CSU actions, with parameters and values which is then written to persistent storage as per the format explained in Fig. 2.

The log files i.e., *LFs* are delivered to a specified location once every configured time interval (eg: once per hour) in compressed format. We examined the size of the files to ascertain possible overhead for providing such services. On the average the record size was around 400 bytes with minimal set of parameters logged (as shown in the result data sets e.g., Fig. 5) and about 1200 bytes of data per activity record with a detailed set of parameters logged. Since the file have been compressed, the storage size decreases approximately by ten fold. (In our environment generating around 40,000 records, consumed about 4.8MB of storage). Therefore the storage cost is insignificant. By applying life cycle rules (i.e., retention period to the logs) and purging the files can reduce the storage even cost further.

4.3. Validation of the framework

In this section we validate the proposed cloud forensic log format, with respect to forensic practitioners need. We validated the log architecture using complex use cases and large set of data. However, we present here two samples and simple use cases, as a proof of validation of the concept.

4.3.1. Use cases

Use Case 1. The cloud storage is commonly used to store large set of confidential personal information like social security or identification card numbers. Business often use cloud to store personal data and credit card details of their clients to enable faster transactions. Hence cloud is an attractive target for data thieves. But for the security engineers and law enforcement agencies the very nature of the cloud computing makes it very difficult to secure and enforce the law and to catch the wrongdoers. In this particular use case we assume that a malicious actor, referred as *(user_hacker)* register for a CSU account, (possibly with an assumed name), legitimately procure reasonably large computing power and services in the cloud. Using the cloud resources the *user_hacker* then executed malicious programs, such as password cracking, harvesting credit card numbers etc. The results were collected in a file. Subsequently the results are shipped out. The *user_hacker* vanished after deleting his account with the cloud and completely terminating the services. When the account gets terminated the storage allocated on cloud infrastructure will be quickly re-allocated to other CSUs, thereby enabling the criminals to disappear with no trace left.

For the Use Case 1, we consider the following test scenario:

1. Register and create CSU account for *user_hacker*
2. Login to the cloud as *user_hacker*
3. Acquire cloud computing resources, such as the computing power, memory and storage.
4. Create folder and storage locations necessary to run the program.
5. Upload malicious programs and files.
6. Execute the malicious programs.
7. Collect the resulting output file.
8. Download the output files.
9. Log out of the session.
10. Finally terminate the CSU account of *user_hacker*

Use Case 2. In this Use Case we assume that a malicious actor *(User_Evil)* trades child pornography over internet. He used to collect and keep large set of pornography image files in his laptop. Later *User_Evil* found out that cloud computing environment is the best way to distribute the files with no traces of actions left behind. Subsequently *User_Evil* created cloud account, acquired the

necessary storage space and uploaded the image files to the cloud storage. The files were then wiped out from the local machine using special tools to erase any possible traces. *User_Evil* used the cloud services as a storage and transmission centre and quickly terminated the account once the job has been done.

For the Use Case 2, we consider the following test scenario:

1. *User_Evil* creates a CSU account and logs in.
2. *User_Evil* upload illegal image files to the cloud
3. The image files are made available to the potential buyers by sharing it. The buyers can then download the files as they wish.
4. *User_Evil* logs out of the session
5. Subsequently *User_Evil* terminates his CSU account

5. Results

We conducted extensive tests using the purpose built own-Cloud experimental environment. Some sample results of the experiments are listed in Appendix - A and shown in Figs. 4–6. Fig. 4 shows the result of an admin user login and creating (or deleting) CSU accounts, which is an equivalent action of CSUs creating account for themselves (or terminating their own account) in a public cloud environment. Fig. 5 provides the result of *Use Case 1* with geographic location recorded. Fig. 6 provides the result of *Use Case 2*.

5.1. Analysis of Results

In this section we provide the analysis of results against ACPO guidelines and NISTIR 8006 report.

5.1.1. Analysis of results against ACPO guidelines

The applicability of ACPO guidelines for conducting cloud forensics has been analyzed by many researchers. The study focused on the impact of ACPO guidelines on the core principles of investigations in cloud computing platform. The study concludes that the ACPO guidelines can in general be applied to cloud computing forensics investigations too, even though the ACPO guidelines does not make any reference to cloud computing as such. However the study recommends to take up additional precautions, in particular while collecting evidence [37,38].

We are using the ACPO guidelines to validate the applicability of this framework as explained below:

a) *Audit Trail*. Principle 3 of the ACPO guidelines says [29]: *"An audit trail or other record of all processes applied to computer based electronic evidence should be created and preserved. An independent third party should be able to examine those processes and achieve the same result."*
This framework records and stores all the CSU activities as a mandatory task. Repeating the activities by any one produces the same result. The audit trail generated is easy to understand and does not require any specific expertise, thereby partially supporting the Principle 2 of the guidelines which emphasizes the competency of the people to understand and interpret the evidence. The framework is scalable and additional parameters can be easily added. The sub-clause 3.5 of the guidelines recommends in relation to mobile phone data collection that it is preferable to collect the call logs from the communication service provider rather than requesting the forensic examination of the mobile phones. Extrapolating the same principle to cloud forensic, we can say that it is preferable to collect the logs from the CSPs, than seizing and examining the client end-point devices. This log framework enables the CSPs to retrieve and provide the logs associated with CSUs very easily. This log framework clearly identifies key parameters like the originator IP address, time and action carried out, and presents sequentially as a chain of events. The framework mandates creating logs per CSU basis, i.e., *LFPUs*, therefore the logs are not co-mingled with other CSUs activities and does not require any processing while making the logs available to the investigators. Thereby strengthening the trust and integrity of the forensic artefacts.

b) *Provenance*. Provenance is the science of associating an evidence to a suspect or potential criminal. It is very important in any investigative forensics, let alone digital forensics. For example investigators tries to capture the finger print or blood traces etc. from a crime scene, to establish the provenance. Digital evidence is no difference either. In this regard the quoting Section 5.10.3 of the ACPO guidelines states; *"Establishing the provenance of digital evidence is another key task of the forensics practitioner, who must use their knowledge and skills to identify not just that the evidence exists but also how it came to be there... ... .It is the responsibility of the practitioner to carry out analysis to identify provenance where necessary, to mitigate the risk of their findings being misinterpreted."* The framework, proposed in this paper supports the provenance requirements too. As described in the cloud forensic log architecture described in 4.1 each log entry captures the key parameters, such as source CSU id, ip address, date, time, action etc., which helps to establish the provenance of digital evidence collected. Also the log structure is an incident recording of events in a logical and structured format and is self explanatory. Therefore to analyze, interpret and to present the facts collected from the logs is rather easy and poses least risk of the findings being misinterpreted.

c) *Simplicity*. The simplicity feature of the log framework as depicted in 4.1, in particular the recommendations to separate the files by CSUs and to store the files in a pre-configured location in CSP controlled area of the cloud stack, makes easier to acquire the evidence for the investigators, and therefore, supports Section 5.10.5 of the ACPO guidelines. The Section 5.10.5 of the guidelines cautions the practitioners to be careful while stepping out of their knowledge boundary and suggests to seek the expertise of additional specialists when necessary.

d) *Evidence acquisition*. The framework further helps to support Section 4.3.2 of the guidelines that deals with evidence seizure which states *"... The person in charge of the search must have reasonable ground to remove property and there must be justifiable reasons for doing so... "*. The log parameters in our proposed CFLOG structure definitely helps the investigators in arriving to a sound and justifiable reasoning regarding further actions related to evidence acquisition.

e) *Interoperability*. Lack of interoperability, including the non-existence of standard format for logging, between cloud service providers has been cited as one of the major cloud forensic challenges by previous researchers [5,9]. The lack interoperability between CSPs further complicates factors such as evidence correlation too. The log framework that we proposed in CFLOG helps to address this problem to a significant level. If the service providers use the proposed format and principles of logging, then it would really simplify the process of collating and unifying logs from different providers, and thereby making much easier to correlate the evidence.

f) *Trust.* Since, the CFLOG application is running at the Hypervisor level,and the generated logs are stored outside the CSU partition and control, adds significantly to increase the confidence in logs, which serves as a prime forensic artefacts.

5.1.2. Analysis of results against NISTIR 8006

NISTIR 8006 identifies and summarizes the Cloud Computing Forensic science challenges [9]. We are listing few of the relevant challenges, related to the logs, analyses those challenges in light of the proposed framework.

a) *Deletion of objects in the cloud*. Attributing deleted data to a specific CSU as well as recovering deleted objects in cloud is a big challenge. Though the logging services provided by CSPs do log such activities, but those services have to be acquired and configured by the CSU, which cannot be expected from a malicious actor. In any case, the logs will be lost, once the client terminates the account completely. In our proposed model *CFLOG*, all such activities are logged mandatory and persisted, thereby providing a vital link to the forensics investigation.
b) *Log Format Unification*. As different CSPs have different architecture, so do they differ in event logging format. However, if the providers accept the proposed uniform structure basically providing, "who, what, where and when", information, it can significantly reduce the overhead when the logs are collated and filtered looking for forensic threads. Following uniform logging structure also would help to ease another challenge i.e., *Interoperability issues among providers* in the logging space.
c) *Time line analysis of logs*. Having time recorded in UTC format, as specified in the Section 4.1 helps and makes it easier to do the forensic analysis and data correlation along a uniform time line.
d) *Detection of the malicious act*. Attacks on the computer systems are carried out in incremental steps, where each step exploits a small vulnerability and can easily go unnoticed until the attacker penetrates the cloud and a major system compromise happens. By logging the activities at Hypervisor level, malicious actors cannot reach to the logs, i.e., the *LFs* are protected. Analysing it systematically and routinely would help to find out any suspicious activity earlier.
e) *Evidence segregation*. Traditionally logs collect all the activity of all the CSUs over a period of time in one (or set of) file(s). When investigators want to trace the activity related to a specific person requires filtering and processing of the log files to extract the evidence of interest. Our model recommends to have log file per user (*LFPU*) as described in Section 4.1, hence partially resolving the evidence segregation issue on the logs.
f) *Locating evidence and E-discovery*. Evidence collection is often infeasible in the cloud, as specific location of evidence are unknown. The proposed log framework can write the log output to a predefined location, which can be made known only to the investigator on demand, further easing e-discovery process and the dependency on cloud service providers.
g) *Selective data acquisition*. Selecting data for acquisition remains to be a challenge, because of the multi-tenancy nature of the cloud. Our log framework proposes recording of actions per CSU. It helps to reduce the overall data set, which the investigator is interested and narrows down to the richest sources of information.
h) *Service Level Agreement (SLA)*. Lack of forensic related terms in SLA has been cited as cloud forensic challenge [4,9]. Our log framework proposes a mandatory recording of actions as explained in Section 4.1. Thereby easing the challenge, despite the existence of forensic friendly SLA.

6. Conclusion

Logs, detailing the CSU actions and events are a formidable part of digital forensics and it is even more important for conducting the digital forensics in the cloud computing. Any computing systems produces numerous logs. Collection and analysis of the logs is an enormous and challenging task, especially in a cloud computing environment. CSPs are the custodian of the data assets and it is challenging for investigators to seize the evidence from cloud computing platform. Therefore investigators have to depend upon CSPs for evidence acquisition. Analysis of the logs is similar to finding out needle in a haystack, requiring specialist tools and expertise to filter out the noise, extract the relevant data and then to connect the chain of events. Any modifications to the original log data can undermine the legal validity of the evidence too. Also, the logs are not much useful if it doesn't contain information that satisfies the legal needs and valid in the eyes of the law.

Many research work has been published in the field of cloud computing logging, primarily in the field of security, transporting, maintaining the integrity and confidentiality of logs. Unfortunately none of them addressed the needs of forensic investigators. In this paper we analysed the needs of the forensic practitioners and proposed a log framework addressing the practitioners needs from a business angle. We used the ACPO guidelines as the primary source of the business needs. We designed and build the application, validated the framework on an ownCloud platform. Further validation of the framework has been carried out using two use case examples and proving the results validated against ACPO principles.

In addition the framework has been validated against NIST draft report on Cloud Computing Forensic science challenges. We established the validity of the framework with adequate reasoning and proof. We also found out that the extra service does not add any significant overhead to the CSPs.

In summary, the framework enables the following forensic activities:

- To re-create the events.
- To trace chain of events and build a corroborative evidence set.
- To easily attribute an action to a CSU.
- To clearly separate the CSU logs.
- To retain the logs, even if the CSU account has been terminated.
- To acquire the logs without affecting other consumers.
- To establish trust and confidence in logs to a significant level.
- To easily interpret the logs.

Therefore, we believe that the framework would significantly advances the efficiency and speed of cloud forensic investigations.

Future work. We plan to develop a process and methodology to enable the investigators directly collect logs without going thru the CSPs, regardless of the jurisdictional boundaries and at the same time ensuring the legal validity of logs.

Appendix A


```
{"Records": [
        {
        "UTC_timestamp": "2016-03-31 03:20:49",
        "user": "admin",
        "src_ip": "134.7.56.1",
        "src_port": "54510",
        "dest_ip": [
                "134.7.57.9"
                ],
        "local_time": "2016-03-31 11:20:49",
        "proto": "http",
        "file_params1": {
                "folder": "",
                "file_name": "",
                "size": ""
                },
        "file_params2": null,
        "action": "User logout"
        },
        {
        "UTC_timestamp": "2016-03-31 08:20:18",
        "user": "admin",
        "src_ip": "134.7.56.1",
        "src_port": "54510",
        "dest_ip": [
                "134.7.57.9"
                ],
        "local_time": "2016-03-31 16:20:18",
        "proto": "http",
        "file_params1": {
                "folder": "",
                "file_name":   "user_hacker",
                "size": ""
                },
        "file_params2": null,
        "action": "user deleted"
        },
        {
        "UTC_timestamp": "2016-03-31 03:18:18",
        "user": "admin",
        "src_ip": "134.7.56.1",
        "src_port": "54472",  "
        "dest_ip": [
                "134.7.57.9"
                ],
        "local_time": "2016-03-31 11:18:18",
        "proto": "http",
        "file_params1": {
                "folder": ".",
                "file_name":   "user_hacker",
                "size": ""
                },
        "file_params2": null,
        "action": "New user created"
        },
        {
        "UTC_timestamp": "2016-03-29 08:18:14",
        "user": "admin",
        "src_ip": "134.7.56.1",
        "src_port": "55083",  "
        "dest_ip": [
                "134.7.57.9"
                ],
        "local_time": "2016-03-29 16:18:14",
        "proto": "http",
        "file_params1": {
                "folder": "",
                "file_name": "",
                "size": ""
        },
        "file_params2": null,
        "action": "User login"
        },
        {
        "UTC_timestamp": "2016-03-29 08:18:14",
        "user": "admin",
        "src_ip": "134.7.56.1",
        "src_port": "55083",
        "dest_ip": [
                "134.7.57.9"
                ],
        "local_time": "2016-03-29 16:18:14",
        "proto": "http",
        "file_params1": {
                "folder": "",
                "file_name": "",
                "size": ""
                },
        "file_params2": null,
        "action": "User login attempt"
        },
        {……..
        },
    ]
}
```


Fig. 4. Logs of administering CSU accounts.

```
{ "Records": [
        {
        "UTC_timestamp": "2016-03-31 07:38:45",
        "user": "user_hacker",
        "src_ip": "134.7.56.1",
        "src_port": "61212",
        "dest_ip": [
                "134.7.57.9"
                ],
        "local_time": "2016-03-31 15:38:45",
        "proto": "http",
         "file_params1": {
                "folder": "",
                "file_name": "",
                "size": “”
                },
        "file_params2": null,
        "action": "User logout"
        },
        {
        "UTC_timestamp": "2016-03-31 07:10:03",
        "user": "user_hacker",
        "src_ip": "134.7.56.1",
        "src_port": "61093",
        "dest_ip": [
                "134.7.57.9"
                ],
        "local_time": "2016-03-31 15:10:03",
        "proto": "http",
        "file_params1": {
                "folder": "\/Documents\/App Storage",
                "file_name":
                "sample_pwd_cracker_output.xml",
                "size": "568000"
               “location”:
               “Perth\Australia\Lat:32.0062\Long:115.8944”
                },
                "file_params2": null,
                "action": "File downloaded"
        },
        {
        "UTC_timestamp": "2016-03-31 07:07:57",
        "user": "user_hacker",
        "src_ip": "134.7.56.1",
        "src_port": "61093",
        "dest_ip": [
                "134.7.57.9"
                ],
        "local_time": "2016-03-31 15:07:57",
        "proto": "http",
        "file_params1": {
                "folder": "\/Documents\/App Storage",
                "file_name":
                "sample_pwd_cracker_output.xml",
                "size": "568000"
                “location”:
                “Perth\Australia\Lat:32.0062\Long:115.8944”
                },
                "file_params2": null,
                "action": "File written"
        },
        {
        "UTC_timestamp": "2016-03-31 06:46:22",
        "user": "user_hacker",
        "src_ip": "134.7.56.1",
        "src_port": "56156",
        "dest_ip": [
                "134.7.57.9"
                ],
        "local_time": "2016-03-31 14:46:22",
        "proto": "http",
        "file_params1": {
                "folder": "\/Documents\/App Storage",
                "file_name":  "sample_pwd_cracker.exe",
                "size": "248000"
                “location”:
                “Perth\Australia\Lat:32.0062\Long:115.8944”
                },
        "file_params2": null,
        "action": "File run"
        },
        {
        "UTC_timestamp": "2016-03-31 06:42:41",
        "user": "user_hacker",
        "src_ip": "134.7.56.1",
        "src_port": "56100",
        "dest_ip": [
                "134.7.57.9"
                ],
        "local_time": "2016-03-31 14:42:41",
        "proto": "http",
        "file_params1": {
                "folder": "\/Documents\/App Storage",
                "file_name":  "sample_pwd_cracker.exe",
                "size": "248000"
                “location”:
                “Perth\Australia\Lat:32.0062\Long:115.8944”
                },
        "file_params2": null,
        "action": "File written"
        },
        {
        "UTC_timestamp": "2016-03-31 03:20:46",
        "user": "user_hacker",
        "src_ip": "134.7.56.1",
        "src_port": "54529",
        "dest_ip": [
                "134.7.57.9"
        ],
        "local_time": "2016-03-31 11:20:46",
        "proto": "http",
        "file_params1": {
                "folder": "",
                "file_name": "",
                "size": “”
                },
        "file_params2": null,
        "action": "User login"
        },
   ]
}
```


Fig. 5. Results of use case 1.

```
{ "Records":[
  {
  "UTC_timestamp":"2016-06-08  14:04:24",
  "user": "User_Evil",
  "src_ip": "134.7.49.113",
  "src_port": "49294",
  "dest_ip": [
                "134.7.57.9"
              ],
  "local_time": "2016-06-08 22:04:24",
  "proto": "http",
  "file_params1": {
        "folder": "",
        "file_name": "",
        "size": ”
         },
     "file_params2": null,
     "action": "User logout"
   },
   {
   "UTC_timestamp":"2016-06-08  14:04:15",
   "user": "User_Evil",
   "src_ip": "134.7.49.113",
   "src_port": "49293",
   "dest_ip": [
                "134.7.57.9"
              ],
   "local_time": "2016-06-08 22:04:15",
   "proto": "http",
   "file_params1": {
         "folder": "\\",
         "file_name": "Desert.jpg",
         "size": ”295752”
         },
    "file_params2": null,
    "action": "File shared"
   },
   {
   "UTC_timestamp":"2016-06-08  14:04:03",
   "user": "User_Evil",
   "src_ip": "134.7.49.113",
   "src_port": "49293",
   "dest_ip": [
                "134.7.57.9"
              ],
   "local_time": "2016-06-08 22:04:03",
   "proto": "http",
   "file_params1": {
        "folder": "\\",
        "file_name": "Penguins.jpg ",
        "size": "1387646"
        },
   "file_params2": null,
   "action": "File shared"
   },
```

```
{
"UTC_timestamp": "2016-06-08 14:01:34",
"user": "User_Evil",
"src_ip": "134.7.49.113",
"src_port": "49278",
"dest_ip": [
              "134.7.57.9"
            ],
 "local_time": "2016-06-08 22:01:34",
 "proto": "http",
 "file_params1": {
      "folder": "\/Documents\/Test_Site"
      "file_name": "Penguins.jpg",
      "size": "1387646"
      },
    "file_params2": null,
    "action": "File written"
 },
 {
 "UTC_timestamp": "2016-06-08 14:00:50",
 "user": "User_Evil",
 "src_ip": "134.7.49.113",
 "src_port": "49272",
 "dest_ip": [
               "134.7.57.9"
             ],
 "local_time": "2016-06-08 22:00:50",
 "proto": "http",
 "file_params1": {
      "folder": "\/Documents\/Test_Site"
      "file_name": "Desert.jpg",
      "size": "295752"
      },
   "file_params2": null,
   "action": "File written"
 },
 {
 "UTC_timestamp": "2016-06-08 13:59:13",
 "user": "User_Evil",
 "src_ip": "134.7.49.113",
 "src_port": "49237",
 "dest_ip": [
               "134.7.57.9"
             ],
 "local_time": "2016-06-08 21:59:13",
 "proto": "http",
 "file_params1": {
      "folder": "",
      "file_name": "",
      "size": false
       },
   "file_params2": null,
   "action": "User login"
  }
  {……..
  },

 ]
}
```


Fig. 6. Results of use case 2.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.jisa.2018.07.008.

References

[1] Deeter B, Shen K. The state of the cloud report 2015. https://www.bvp.com/blog/state-cloud-2015 [Accessed:May 2017], 2015.
[2] Deeter B, Shen K, Khan A. The state of the cloud industry 2018. https://www.bvp.com/blog/state-cloud-industry-2018 [Accessed: May 2018], 2018.
[3] RightScale. State of the cloud report 2017. http://www.rightscale.com/2017-cloud-report [Accessed: May 2017], 2016.
[4] Ruan K, Carthy J, Kechadi T, Crosbie M. Cloud forensics. In: Advances in digital forensics VII. Springer; 2011. p. 35–46.
[5] Pichan A, Lazarescu M, Soh ST. Cloud forensics: technical challenges, solutions and comparative analysis. Digital Invest 2015;13:38–57.
[6] Taylor M, Haggerty J, Gresty D, Lamb D. Forensic investigation of cloud computing systems. Network Secur 2011;2011(3):4–10.
[7] Grispos G, Storer T, Glisson WB. Calm before the storm: the challenges of cloud. Emerging Digital Forensics Appl Crime Detect, Prevent, Secur 2013;4:28–48.
[8] Raju B, Moharil B, Geethakumari G. FaaSeC: enabling forensics-as-a-service for cloud computing systems. In: Proceedings of the 9th international conference on utility and cloud computing. ACM; 2016. p. 220–7.
[9] Mell P, Grance T. NIST cloud computing forensic science challenges. Draft Nistir 2014;8006.
[10] Zawoad S, Dutta AK, Hasan R. Towards building forensics enabled cloud through secure logging-as-a-service. IEEE Trans Dependable Secure Comput 2016;13(2):148–62. doi:10.1109/TDSC.2015.2482484.
[11] Goodin D. Amazon cloud hosts nasty banking trojan. http://www.theregister.co.uk/2011/07/29/amazon_hosts_spyeye/ [Accessed: April 2017], 2011.
[12] Sang T. A log based approach to make digital forensics easier on cloud computing. In: Intelligent system design and engineering applications (ISDEA), 2013 third international conference on; 2013. p. 91–4. doi:10.1109/ISDEA.2012.29.
[13] Holt JE. Logcrypt: forward security and public verification for secure audit logs. In: Proceedings of the 2006 Australasian workshops on grid computing and e-research-Volume 54. Australian Computer Society, Inc.; 2006. p. 203–11.
[14] Marty R. Cloud application logging for forensics. In: Proceedings of the 2011 ACM symposium on applied computing. ACM; 2011. p. 178–84.
[15] Patrascu A, Patriciu V-V. Logging system for cloud computing forensic environments. J Control Eng Appl Inf 2014;16(1):80–8.
[16] Sang T. A log based approach to make digital forensics easier on cloud computing. In: Intelligent system design and engineering applications (ISDEA), 2013 third international conference on. IEEE; 2013. p. 91–4.
[17] Jansen W, Grance T, et al. Guidelines on security and privacy in public cloud computing. NIST special publication 2011;800(144):10–11.
[18] Kent K, Chevalier S, Grance T, Dang H. Guide to integrating forensics techniques into incident response, national institute of standards and technology NIST special publication (sp) 800-86. Computer Security Division, Information Technology Laboratory, Gaithersburg, MD; 2006. http://csrc.nist.gov/publications/nistpubs/800-86/SP800-86.pdf.
[19] Reilly D, Wren C, Berry T. Cloud computing: pros and cons for computer forensic investigations. Int J multimedia image process (IJMIP) 2011;1(1):26–34.
[20] Zafarullah Z, Anwar F, Anwar Z. Digital forensics for eucalyptus. In: Frontiers of information technology (FIT), 2011; 2011. p. 110–16. doi:10.1109/FIT.2011.28.
[21] Dykstra J, Sherman AT. Acquiring forensic evidence from infrastructure-as-a-service cloud computing: exploring and evaluating tools, trust, and techniques. Digital Invest 2012;9:S90–8.
[22] Dykstra J, Sherman A. Design and implementation of FROST: digital forensic tools for the openstack cloud computing platform. Digital Invest 2013;10:S87–95.
[23] Zawoad S, Dutta AK, Hasan R. Seclaas: secure logging-as-a-service for cloud forensics. In: Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security. ACM; 2013. p. 219–30.
[24] Schneier B, Kelsey J. Secure audit logs to support computer forensics. ACM Trans Inf Syst Secur (TISSEC) 1999;2(2):159–76.
[25] Battistoni R, Di Pietro R, Lombardi F. CURE-Towards enforcing a reliable timeline for cloud forensics: Model, architecture, and experiments. Comput Commun 2016;91:29–43.
[26] Ruan K, Carthy J, Kechadi T, Baggili I. Cloud forensics definitions and critical criteria for cloud forensic capability: an overview of survey results. Digital Invest 2013;10(1):34–43.
[27] AWS. AWS CloudTrail. https://aws.amazon.com/cloudtrail/, 2018.
[28] Azure. Azure monitoring and diagnostics. https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-overview-activity-logs, 2018.
[29] ACPO. ACPO good practice guide for digital evidence (Version 5.0). http://www.digital-detective.net/digital-forensics-documents/ACPO_Good_Practice_Guide_for_Digital_Evidence_v5.pdf, 2012.
[30] Adams R. The emergence of cloud storage and the need for a new digital forensic process model. Cybercrime Cloud Forens 2012:79.
[31] Bohn RB, Messina J, Liu F, Tong J, Mao J. Nist cloud computing reference architecture. In: Services (SERVICES), 2011 IEEE world congress on. IEEE; 2011. p. 594–6.
[32] Ab Rahman NH, Glisson WB, Yang Y, Choo K-KR. Forensic-by-design framework for cyber-physical cloud systems. IEEE Cloud Comput 2016;3(1):50–9.
[33] Ab Rahman NH, Cahyani NDW, Choo K-KR. Cloud incident handling and forensic-by-design: cloud storage as a case study. Concurrency Comput 2017;29(14).
[34] Martini B, Choo K-KR. Cloud storage forensics: owncloud as a case study. Digital Invest 2013;10(4):287–99.
[35] Alex ME, Kishore R. Forensics framework for cloud computing. Comput Electr Eng 2017;60:193–205.
[36] Ab Rahman NH, Choo K-KR, et al. Integrating digital forensic practices in cloud incident handling: a conceptual cloud incident handling model. In: The cloud security ecosystem: technical, legal, business and management issues. Syngress Publishing; 2015. p. 383–400. doi:10.1016/B978-0-12-801595-7.00017-3.
[37] Lallie HS, Pimlott L. Applying the ACPO principles in public cloud forensic investigations. J Digital Forens, Secur Law 2012;7(1):71.
[38] Lallie HS. Challenges in applying the ACPO principles in cloud forensic investigations. J Digital Forens, Secur Law 2012;7(1):71–86.