

Cloud Log Forensics: Foundations, State-of-the-art, and Future Directions

SULEMAN KHAN, University of Malaya, Malaysia
ABDULLAH GANI, University of Malaya, Malaysia
AINUDDIN WAHID ABDUL WAHAB, University of Malaya, Malaysia
MUSTAPHA AMINU BAGIWA, University of Malaya, Malaysia
MUHAMMAD SHIRAZ, Federal Urdu University, Islamabad, Pakistan
SAMEE U. KHAN, North Dakota State University, USA
RAJKUMAR BUYYA, University of Melbourne, Australia
ALBERT Y. ZOMAYA, University of Sydney, Australia

Cloud log forensics (CLF) mitigates the investigation process by identifying the malicious behavior of attackers through profound cloud log analysis. However, the accessibility attributes of cloud logs obstructs accomplishment of the goal to investigate cloud logs for various susceptibilities. Accessibility involves the issues of cloud log access, selection of proper cloud log file, cloud log data integrity, and trustworthiness of cloud logs. Therefore, forensic investigators of cloud log files are dependent on cloud service providers (CSPs) to get access of different cloud logs. Accessing cloud logs from outside the cloud without depending on the CSP is a challenging research area; whereas, the increase in cloud attacks has increased the need for CLF to investigate the malicious activities of attackers. This paper reviews the state-of-the-art of CLF and highlights different challenges and issues involved in investigating cloud log data. The logging mode, the importance of CLF, and cloud log-as-a-service are introduced. Moreover, case studies related to CLF are explained to highlight the practical implementation of cloud log investigation for analyzing malicious behaviors. The CLF security requirements, vulnerability points, and challenges are identified to tolerate different cloud log susceptibilities. We identify and introduce challenges and future directions to highlight open research areas of CLF for motivating investigators, academicians, and researchers to investigate on them.

Categories and Subject Descriptors: **H.2.0 [General]**: Security, Integrity, and Protection; **H.2.7 [Database Administration]**: Logging and Recovery

General Terms: Forensics, Reliability, Log Management

Additional Key Words and Phrases: Cloud computing, Cloud log forensics, Big data.

ACM Reference Format:

Suleman Khan, Abdullah Gani, Ainuddin Wahid Abdul Wahab, Mustapha Aminu Bagiwa, Muhammad Shiraz, Samee U. Khan, Rajkumar Buyya, and Albert Y. Zomaya, 2016. Cloud log forensics: Foundations, State-of-the-art, and Future Directions. *ACM Comput. Surv.* 0, 0. Article 0 (2016), 44 pages. DOI:<http://dx.doi.org/10.1145/0000000.0000000>

Author's addresses: S. Khan, A. Gani, A.W.A. Wahab, M.A. Bagiwa, Center For Mobile Cloud Computing Research, University of Malaya, Kuala Lumpur, Malaysia; M. Shiraz, Department of Computer Science, Federal Urdu University of Arts, Science and Technology Islamabad, Pakistan; Samee U. Khan, Department of Electrical and Computer Engineering, North Dakota State University, Fargo, ND 58108-6050 USA; R. Buyya, Department of Computing and Information Systems, The University of Melbourne, Cloud Computing and Distributed Systems Lab, Australia; A.Y. Zomaya, School of Information Technologies, Building J12, The University of Sydney, NSW, Sydney, 2006.

Corresponding author: Abdullah Gani, Center for Mobile Cloud Computing Research (C4MCCR), Faculty of Computer Science and Information Technology, University of Malaya, Malaysia.

Permission to make digital or hardcopies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credits permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2010 ACM 1539-9087/2010/03-ART39 \$15.00

DOI:<http://dx.doi.org/10.1145/0000000.0000000>

1. INTRODUCTION

Any event occurring in an organization IT system or network is recorded with various entries in a log file. The process of recording log files is known as logging [Chuvakin *et al.* 2013]. The log file provides useful information regarding previous events occurring in the system and network during a specified time span. For instance, a network administrator can find out about the network bandwidth usage in a time interval by analyzing the network logs. Similarly, application developers use application logs to identify and fix bugs inside a program code. Each entry in the log file provides significant information related to a particular event at the time the log file is generated. Initially, the log file is used for trouble shooting [Flegel 2002]. Now, the log file provides more functional services including system and network monitoring, optimizing performance of system and network, recording user activity, and investigating malicious behavior [Kent and Souppaya 2014]. Logs are now mainly used for security purposes due to increased attacks on the system and network [Zuk 2011]. The logs used to record attackers activities at the time of the attack, help system and network administrators to investigate attacks by analyzing log file data [Mao *et al.* 2014].

In large organizations, different types of log files are created on different devices that involve the issue of effective management of logs due to scarcity of resources. To overcome the log management problem, organizations have started to move towards cloud computing by using cloud logging services known as log-as-a-service [Saurabh and Beedgen 2014]. Log files generated on different organizational resources are sent to the cloud for storage and analysis using cloud storage resources and cloud log analysis servers. Similarly, organizations mainly run their applications in computational clouds that also require logging to investigate malicious activities when detected. Cloud logging includes cloud application logs, cloud network logs, cloud system logs, cloud firewall logs and others. In this paper, the phrase cloud log is used to refer to all logs created within a cloud computing environment. Nowadays, attacks on cloud computing are occurring more frequently, which worry users and organizations about how to keep their data safe from different attackers [Khan *et al.* 2014]. Cloud log files record different events occurring in the system and network and are used to investigate different attacks [Vrable *et al.* 2012]. A suitable option is to search the cloud log files for malicious behavior by analyzing them using log analysis methods [Lin *et al.* 2013, Wei *et al.* 2011]. The process of analyzing cloud log files in cloud computing or through third party analysis services is called cloud log forensics (CLF) [Thorpe *et al.* 2012].

CLF is a new emerging field of data security used to analyze data inside cloud log files for the investigation of malicious behavior. However, cloud log files are only accessible to a Cloud Service Provider (CSP) through cloud resource ownership. For instance, in cloud computing Software-as-a-Services (SaaS), user is provided with developed software to run its applications. Each application generates log files during its execution on the cloud that are inaccessible to the users [Ruan *et al.* 2011]. Although cloud log files are not directly accessible to the investigator, the CSPs provide access to such log files with legal approval from the court. CSPs provide restricted access to third party investigators for cloud log files due to user data privacy and organizational Standard Operating Procedures (SOPs) [Birk and Wegener 2011]. Moreover, CLF adopts similar general procedural steps to digital forensics such as the collection, preservation, analysis, and reporting [Khan *et al.* 2014, Sang 2013]. In the collection step, cloud log files are retrieved from different

cloud resources. Different cloud log files collected from different cloud resources may differ depending on the organizational requirements for the cloud log data which include a number of log entries, log file limit, time to log data, and what content to log. After collection, cloud log files are stored in a secure manner to protect the integrity. Data integrity is preserved in CLF for the reason to provide evidence against attackers in the court [Joo *et al.* 2014]. The next step is to perform analysis of the cloud log files to produce potential evidence to help the investigator to track the attacker by re-generating the malicious activities again. The analysis performed on cloud logs provides a clear picture of the malicious activity performed by the attacker during the attack. Cloud log file analysis is the backbone of CLF identifying attacks and assisting administrators to prevent similar types of attacks in the future. Finally, after the analysis performed on cloud logs a legal report is generated to record each event performed during individual steps of the CLF. The report contains comprehensive information regarding entire investigation process, but some of the information includes when the investigation was performed, what was the procedure to collect the evidence, how the integrity of cloud log files was kept, what analysis tools were used, and various others. Usually, the final report is used against the attacker in a court for its malicious behavior.

Moreover, in the past few decades, cloud computing was considered a secure place to store and compute data of different users and organizations. Currently, exploitation of different cloud resources, applications, network channels, and log data have shown that various vulnerabilities are found in cloud computing. To minimize the vulnerabilities found in cloud computing, CSPs start re-thinking for their security matters. The CLF is one aspect of cloud security which assists CSP with in-depth understandability regarding the steps performed in cloud log attacks. The significance of CLF increases when cloud log files store in cloud computing become victims through various attacks include modifying of log data in log files, deleting log data and log files, inserting spoofed log data, and others. The CLF performs deep inspection of infected cloud log files to understand the suspicious behavior of the attack performed on cloud log files. The ultimate goal of CLF is to identify the root cause of the cloud log attacks which helps CSPs to prevent such attacks from repeating again.

The goal of this survey is to provide insight about CLF and to provide researchers with an in-depth understanding through log management [Ray *et al.* 2013], logging modes [Rafael 2013], services of cloud computing log-as-a-service vendors [Ellis 2013; Burton 2014; [IBM 2014, Logentries 2014, Williams 2013], and especially CLF case studies [South 2013] [Beaver 2015]. Moreover, CLF challenges are identified to help researchers in exploring new research areas and motivating them to come up with new ideas, methods, standards, and tools for the advancement of log investigation in cloud computing. To the best of our knowledge, this survey can be considered unique, as no single survey is available on CLF to date. The **key contributions** of this article are highlighted as under:

- Comprehensive background knowledge of CLF: we provide information regarding logging, including its types and logging mode, cloud computing, and digital forensics.
- A brief description of the log-as-a-service provided by cloud vendors: we provide knowledge about how and what features are provided by cloud vendors to their customers regarding cloud log management.

- An explanation of the practical implementation of CLF through case studies: we highlight real world scenarios related to clients and cloud log vendors in deployment and implementation of CLF.
- The identification of CLF security requirements, vulnerability points, and state-of-the-art challenges: we discussed what should be key security parameters for CLF, where should we collect evidence for the investigation, and what are the key challenges for CLF in current.
- Introducing future research directions: we provide potential research areas for CLF to work on it by overcoming its current challenges.

The rest of the paper is organized as follows. Section 2 provides background knowledge of logging by giving an overview of its types and modes. In addition, brief descriptions about cloud computing and digital forensics are provided to gain insight about its core concept. In Section 3, we present importance of CLF and explain state-of-the-art in current. Section 4 explains different cloud vendors that provide log-as-a-service. In Section 5, we describe different case studies related to CLF. The Section 6 introduces CLF security requirements, vulnerability points, and state-of-the-art challenges. Lastly, Section 7 concludes the paper with highlighting future research directions.

2. BACKGROUND

2.1 Logging

The process of recording events in a file during the execution of the operating system, process, system, network, virtual machine, or application is called logging and the file is called a log file [Kent and Souppaya 2014]. The log file contains the sequential steps performed during an execution along a specified timeline. A log file is composed of log entries and each log entry contains useful information associated with events occurred in the system, network, virtual machine, or application. A log file entries differ with its types and requirements. For instance, the standard format used by the web-server to generate server log files includes *[host ident authuser date request status bytes]*. The *'host'* is the client that makes a request to the web-server, *'ident'* is RFC 1413 identifier of the client, *'authuser'* is the user-id used in the request for a document, *'date'* is the date, time, and time-zone field when the webserver finishes the processing of a request, *'request'* is the method requested by the client, *'status'* represents an HTTP status code, and *'bytes'* is the size of an object return to the client by the web-server. For a clear understanding of the log format, Figure 1 depicts to illustrate an access log format with highlighting its different fields. Each log field with its value and description is shown in the Table 1. Each organization has different purposes to generate log files depending on its requirements. Log files are initially generated within organizations for the purpose of troubleshooting; however, the objectives are expanded to many other purposes, including the recording of user actions, user authentication, network performance, optimization, system health monitoring, privacy of data, forensic, and others.

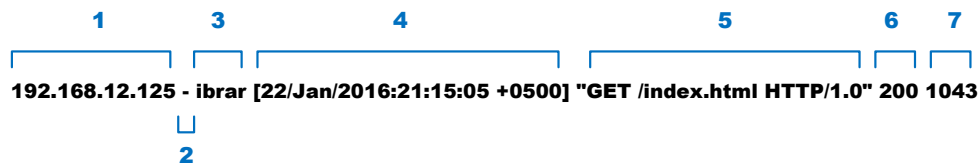


Fig. 1. Format of an access log file

Logging is considered an essential means of security control which helps investigators in identifying, answering, and precluding operational issues, incidents, violations, and fraudulent activities [Kent and Souppaya 2014]. Logging is mainly used in monitoring systems to collect data for investigating different malicious attacks. The logs help investigators to identify the sources of messages generated from various devices at different time intervals. Many logs generated for security reasons stop future intrusions by detecting them through the various patterns and occurrences observed. Audit logs are generated to track user authentication made to the system or network [Prasad and Chakrabarti 2014]. Similarly, security devices, such as intrusion detection systems and firewalls record logs to contain possible attacks [Vaarandi and Pihelgas 2014]. Therefore, different logs can be used for security purposes depending on the organizational requirements. Some security logs are generated on a real-time basis by collecting events during the execution time of the system and network, while some security logs are generated periodically at regular time intervals.

Table 1: Description of the access log format

S.No	Fields	Value	Description
1	host	192.168.12.125	IP address of the HTTP user which makes HTTP resource request
2	rfc931	-	Identifier used to determine the client
3	username	ibrar	User name or User id used for authentication
4	date:time timezone	[22/Jan/2016:21:15:05 +0500]	Date and time stamp of the HTTP request
5	request	"GET /index.html HTTP/1.0"	HTTP request containing (a) HTTP method=GET, (b) HTTP request resource=index.html, and (c) HTTP protocol version= 1.0
6	statuscode	200	Numeric code used to tell about the status of HTTP request i.e. success or failure
7	bytes	1043	Numeric field used to highlight number of bytes of data transferred during the HTTP request

There are several laws and regulations which provide comprehensive guidelines to assist organizations in log management. The Federal Information Security Management Act of 2002 (FISMA) in USA emphasizes that each federal agency should have security measures for their information system infrastructures. The document "NIST SP 800-53" provided by FISMA indicates several controls required for log management, such as log generation, log protection, log retention, and important actions required at the time of audit failure [Force and Initiative 2013]. The document "NIST SP 800-66" provided by the Health Insurance Portability and Accountability Act of 1996 (HIPAA) explains log management by focusing on the need to regularly review access reports and audit logs [Hash *et al.* 2008]. The HIPAA Act emphasizes the need to keep records for every activity and action performed in an organization for up to at least six years. The Payment Card Industry Data Security Standard (PCI DSS) is applied to ensure organizations keep records for credit card holders [Bradley and Dent 2010]. The PCI DSS ensures organizations keep track of all network accessed resources and card holder data information. Similarly, the Gramm-Leach-Bliley Act (GLBA) requires financial institutions to provide security for users' data by providing the proper log management [Janger and Schwartz 2001]. Log management can easily identify violations and vulnerabilities created by the intruders internally or externally in an organization.

2.1.1 Types of Logs

Increasing vulnerabilities, attacks, and violations of organizational data force security personnel to generate different kinds of logs. Every part of a system, application, device, or network which communicates with users or systems need to record communication events in a log file. Examples of various logs include application logs, system logs, security logs, setup logs, network logs, web-server logs, audit logs, VM logs and others. Each of aforementioned log types is briefly described in Table 2 with examples.

The application logs are created by the developers through inserting events in the program. Application logs assist system administrators to know about the situation of an application running on the server. Application logs should be well-structured, event stream, and have relevant information to provide foundations for higher level of abstraction, visualization, and aggregation. The event stream of application logs is necessary to view and filter data coming from multiple instances in the application. The system log files are found in the operating system used to log warning, errors, modify, and debug messages. For instance, a warning message to *“update the device driver”* is recorded in the system logs. The system log files usually contain information regarding data and time of the log creation, type of messages such as debug, error, etc., system generated message related to the occurrence, and processes that have been affected by the occurrence of an event. The security logs are used to provide adequate capabilities in determining malicious activities after their occurrence to prevent them from re-appearing again. Security logs record various information pre-defined initially by the security administrators. For instance, firewall logs provide information related to source routed packets, rejected IP addresses, outbound activities from internal servers, and unsuccessful logins. Security logs provide in-depth information which has to be managed, controlled, and analyzed by the security administrators according to their requirements. The setup log files, records each event during the time of an installation. It assists network administrator to know the sequential steps performed during the installation process which might be useful when there are installation problems. The setup log files generate detail summary regarding installation steps that assist system administrators to follow up the problem easily.

The network log contains detailed information related to different events happened on the network. The event include recording malicious traffic, increase rate of network traffic, packet drops, bandwidth delays, and others. Network administrator monitor and troubleshoot daily networking through analyzing network logs for different intrusion attempts. There are different network devices from which network logs can be collected including routers, network and host-based firewalls, and intrusion detection systems. The web-server logs record entries related to the web pages running on the web-server. The entries contain history for a page request, client IP address, data and time, HTTP code, and bytes served for the request. The web-server logs are accessible to the administrator or webmaster which can perform statistical analysis to find traffic patterns for a specific time interval. The audit log files record unauthorized access to the system or network in a sequential order. It assists security administrators to analyze malicious activities at the time of the attack. Usually, the main information inside audit log files includes source and destination addresses, user login information, and timestamp. The VM log files record information specific to instances running on the VM such as startup configuration, operations, and the time it finishes its execution. VM logs record

different operations i.e. number of instances running on VM, execution time of each application, and application migration to assist CSP in finding malicious activities happens during the attack.

The increasing number of various kinds of logs creates problems for organizations to collect, store, preserve, and analyze log data within the existing infrastructure. The problems faced by organizations in managing log data include human experts, time, cost, tools, resources, and their management. There are lots of difficulties for organizations to build new infrastructure, develop tools, and train their manpower to manage the massive amounts of logs. As a result, higher costs and greater time consumption are required to manage log files with huge amounts of log data.

Table 2: Different types of logs

Types of log	Description	Examples
Application log	Logs that are recorded by an application or program. Application developers are responsible to specify what, when, and how to log through an application execution on a system.	Web applications, Database programs.
System log	System logs are generated by an operating system which are pre-defined and contain information regarding system events, operation, drivers, device change, and various more.	Syslog-ng, Log & Event Manager
Security log	Logs contain security related information to determine malicious behavior found in the system or network. For instance, malware detection, file quarantines, time of malicious detection, and various others.	Event Log Analyzer, Control case Security Event Logging and Monitoring services
Setup log	Setup logs capture the events occur during performing the installation of an application.	Msiexec.exe
Network log	Network log is a log file that contains network related events, i.e. description of the event, priority, time occurrence and much more.	Splunk, Log4j2
Web-server log	Web-server log records all events occur on the web-server such as access time, IP address, date & time, request method, and object volume (bytes).	Nihuo Web Log Analyzer
Audit log	Audit log contains user unauthorized access to the system and network for inspecting its responsibilities. It includes destination addresses, user login information, and timestamp.	WP Security Audit Log, auditpol.exe
Virtual machine logs	A file that contains records of each event performed on a virtual machine.	Virtual Machine Log Auditor, JVM controller

2.1.2 Logging Modes

Logging is the process of recording an event at the time of system execution. When a system is executing correctly, logging creates an overhead of collecting and storing various events in the memory. However, generating logs make a sense when the system goes to the failure stage frequently or various susceptibilities affect the processes in the system. To investigate such problems, logs are required to identify sequential steps of the susceptibilities. There are two main logging modes which specify how the logs should be stored in memory and what should be recovered from logs to investigate different vulnerabilities. Each of the logging modes is briefly explained and the pros and cons of each logging mode are illustrated in Table 3 with their comparison in Table 4.

2.1.2.1 Circular Logging

Circular log refers to the presence of the log in a circular form. Different events are stored in the form of a circular log file that has a pre-defined allocated memory equal to the available memory of the system as shown in figure 2. Each log entry is stored in sequential order in the memory and once the memory reaches its end, the first log entry is automatically overwritten by the newly created log [Wyatt 2009]. The process continues like a revolving ring type. There is no fear that collected logs will overflow the finite memory space. Circular logs are used to restart recovery by rolling back the operational transaction due to the system failure. The queue manager is restarted by accessing the log file without losing the data. During the restart process, log files are acquired against queue files to re-create the transaction message. The reuse of log files for recovery is done through checkpointing [Khan *et al.* 2012]. Checkpointing produces synchronization between queue data and log files to create a point of consistency [Scales *et al.* 2013]. The checkpoint indicates a point where both log file and queue data have the same records at the same time. Therefore, circular logs have less administrative overhead in terms of reduced human intervention. All logs are automatically managed in a pre-defined finite memory without the need for extra memory for the extended log files. The automatic management of log files saves time by reducing the insertion, deletion, and archiving of logs, which speeds up the process with high throughput. However, the overwriting of existing data causes previously stored records, logs to be lost, which might affect the overall recovery process. The log files in circular logging are not archived for long term storage due to their ring type finite memory allocation.

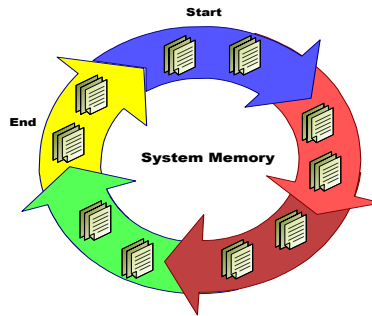


Fig. 2. Generalized circular logging diagram

Table 3: Logging mode advantages and disadvantages

Logging Mode	Advantages	Disadvantages
Circular logging	<ul style="list-style-type: none"> Transaction recovery No maintenance required Applicable for software, power, and application failure Requires minimum human intervention Reused logs Faster throughput No time require for allocation, formation, deletion, and achieving logs 	<ul style="list-style-type: none"> Lack of long term storage Overwrite existing logs by filling finite space No recovery for damage queue files
Linear logging	<ul style="list-style-type: none"> Media recovery Applicable for software, power, application failure, and media failure 	<ul style="list-style-type: none"> Require maintenance Slow process Never reused logs

	<ul style="list-style-type: none"> • Long term storage • Recover damage queue files 	<ul style="list-style-type: none"> • Degrade performance due to periodic allocation of new logs
--	---	--

2.1.2.2 Linear Logging

Linear logging is the process of storing logs in a linear sequential memory space [Turnbull 2005]. The recovery process is the same as for the circular log with more added services such as queue manager, which restarts the process in case of a damaged queue file. The linear log has no finite memory space while its limit is directly proportional to the system’s memory as shown in figure 3. Linear logging stores logs in sequential order inside a memory without overwriting the previous logs [Wyatt 2009]. When the memory is full, previous logs are moved to another memory or they are deleted by the administrator depending on the situation. The memory has no limit for storing logs; it depends on the available capacity of the memory. The linear log stores transaction events as well as a copy of persistent messages. The persistency is a property of a message used to store a message on a disk, database, or to a log file. The persistent message is recovered even the queue manager is restarted. Linear logging recovers the queue files by replaying linear logs, which is also known as media recovery. Therefore, the linear log performs both transaction recovery [On *et al.* 2012] and queue recovery. Transaction recovery is performed by using the checkpoint and queue recovery is performed by using a copy of the persistent message. The linear log has the advantage of using logs for long term storage which is used for analysis whenever it is required. However, linear logs entail maintenance to shift logs from one memory to another storage device when the current memory reaches the peak. The shifting of log files slows down the process and decreases the performance by periodically allocating logs.

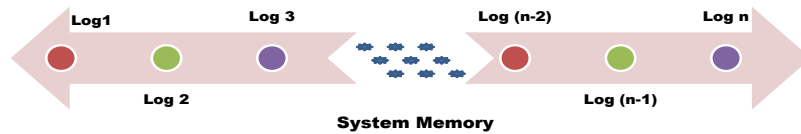


Fig. 3. Generalized linear logging diagram

It is noteworthy to mention that selecting an appropriate logging mode requires an overview of the current requirements. Based on need, one can adopt a logging mode, which should fulfill the requirements of the enterprise. Circular logging performs automatic logs with high performance, whereas, sacrifices the recovery of persistent messages from a damaged queue file. Nevertheless, in the case of linear logging, disk space has to be appropriately managed so that it does not consume all available space. Based on the aforementioned discussion, one has to evaluate each of logging modes based on the cost and risk before their implementation.

Table 4: Comparison between different logging modes

Comparison Parameters	Circular Logging	Linear Logging
Allocation of logs	Once	Periodically
Administrative Overhead	Less (Negligible)	More
Reusability	Yes	No (Logs are moved or deleted)

Comparison Parameters	Circular Logging	Linear Logging
Restart Recovery	Yes	Yes
Recreation of loss data	No	Yes (Replaying logs)
Overwrites log data	Yes	No
Log allocation capacity	Finite	Dynamic

2.2 Cloud Computing

Cloud computing is a connected network resource for providing various services to the users using an Internet communication at any place and time [Armbrust *et al.* 2010; Gani *et al.* 2014; Qi *et al.* 2014]. The resources in the cloud owned or rented out by CSP are integrated together to strengthen the ability of computation and storage [Buyya *et al.* 2008]. The CSP is a company which provides different services to the users by giving access to the cloud resources. Users access cloud resources without having in-depth knowledge or details of its location and ownership. The users are only charged on the basis of cloud resource utilization and such phenomenon is known as “*pay-as-you-go*” in the cloud computing [Armbrust *et al.* 2010]. One resource can be used by many users to increase efficiency, throughput, and also reduce the idle time of the resources in cloud computing.

Moreover, nowadays there are hundreds of CSPs providing different services to users based on their needs, for instance, Microsoft, Amazon, Azure, Google and various others. These CSPs can be categorized into three main service categories which is also known as a service models for cloud computing such as: (a) Infrastructure-as-a-service (IaaS), (b) Platform-as-a-service (PaaS), and (c) Software-as-a-service (SaaS) [Armbrust *et al.* 2010]. In IaaS model, the user is given access to the virtual resources of cloud computing for executing their application while responsible for its application in terms of security, maintenance, and support [Mell and Grance 2011]. Examples include Amazon Web Service (AWS), Google Compute Engine (GCE), Rackspace, and Microsoft Azure. The PaaS model is used by developers to develop new applications on infrastructure provided by the CSPs. In PaaS, CSP assists programmers/developers by providing open/proprietary languages, initial basic configuration for communication, monitoring, distributing the application, scalability of an application, and others [Buyya *et al.* 2008]. The examples for PaaS include AWS Elastic Beanstalk, Force.com, Apprenda, and Heroku. However, in SaaS, CSP provides complete software to the users for its execution. The software/application is accessed through web portal or service oriented architecture [Buyya *et al.* 2009]. Users can access any software listed by CSP without concern about its configuration and installation. The examples of SaaS include Google apps, Gmail, Microsoft 365, Salesforce, and Cisco WebEx. Moreover, other services are also provided by CSP to entertain users to fulfill their requirements through using cloud resources. Some of the services provided by the CSPs are listed in Table 5. Many of the CSPs have now started providing log-as-a-service to their customers by collecting all types of log data [Ellis 2013; Burton 2014; Oppenheimer 2009; Lindvall 2014]. The log data generated in different applications, servers, devices, and networks is normalized and filtered for reformatting before further processing. The log data collected from different organizations are analyzed on cloud resources for different investigative objectives. Cloud log analysis provides useful information to customers including data integration, instant log visibility, real-time monitoring, customize log format, easy and simple diagnosing with trouble shooting, rich GUI features, root cause analysis, and others.

Table 5: Cloud vendors providing different services

Cloud Services	Description	Cloud Vendors
Storage-as-a-Services(S'TaaS)	Provides a huge amount of storage on the cloud architecture to different organization to archive their data. It provides economy of scale and cost reduction benefits in terms of storage as comparative to local available storages.	Amazon S3, Windows Azure Storage
Networking-as-a-Services (NaaS)	To optimize resources by delivering network services through using its transport services. It may provide network virtual services to different users integrated with other service models.	Pertino
Everything-as-a-Services (XaaS)	A group of services deliver through an internet on the cloud infrastructure. For instance, a CSP provides services for logging, storage, forensics, etc.	Google, Microsoft, Hewlett Packard
BigData-as-a-Services (BDaaS)	To deliver statistical analysis tools or information to assist organizations in understanding the large information set to gain competitive advantages.	1010data, IBM, AWS
Forensics-as-a-a-Services (FaaS)	Investigate various cyber-criminal events while using high analytical investigation tools integrated with high performance computing resources.	No specialized vendor available yet
Desktop-as-a-Services (DaaS)	The offering of virtual desktop interface with multi-tenant architecture in a cloud through monthly fee subscription.	Wipro, Citrix XenDesktop
Graphic-as-a-Services (GaaS)	Provides cloud based graphical technologies to run high end graphic design application using HTML5 web-browser.	NVIDIA
Testing-as-a-Services (TaaS)	A testing activities related to the organization are performed in the cloud rather than conducted by employees in the job space.	Oracle, Cognizant

2.3 Digital Forensics

Digital forensics is the process to identify digital artifacts for investigating malicious behaviors of the attacker [Chung *et al.* 2012]. Malicious behaviors of the attacker compromise secret credentials of the user by exploiting its privacy through monitoring, altering, deleting, and copying data on different devices [Casey 2009]. The origin of attackers has to be investigated to prevent malicious behaviors from exploiting legitimate user data. Several digital forensics process models have been proposed to perform digital investigations in different research aspect that includes military, business, law enforcement, and various industries. Nevertheless, different researchers have proposed different digital forensics models, however, National Institute of Standard and Technology (NIST) has explained four general phases of digital forensics in their report [Kent *et al.* 2006] such as collection, examination, analysis, and reporting.

The collection phase is the initial stage of digital forensics in which digital evidence is collected from digital artifacts. This phase is vital in terms of collecting appropriate evidence, however; incorrect acquisition of evidence will bias rest of the digital forensics process. In examination phase, usually massive amount of collected data is processed to identify forensically sound data which have to be investigated for valuable evidences. The integrity of the data has to be preserved by keeping its originality. The analysis phase is used to analyze data to identify various susceptibilities and malicious behaviors of the attacker in the preserve data collected from the examination phase to determine the root cause of the attack. In most of the case, live analysis is required to overcome the intensity of the malicious behavior by identifying the root cause of the attack quickly [Carrier 2006]. The well-known digital forensics tools such as Sleuth Kit, Encase, and Forensic Toolkit (FTK) are

used to identify evidence extracted from the register, temporary, and deleted files as well as email, cache, cookies, and metadata presented in various devices. Finally, in reporting phase the results of the analysis phase are compiled in a shape of legal document which has to be presented in the court against the attacker. The report contains information regarding the method used for the analysis, selection of tools and procedures, necessary actions taken in each phase of the investigation, recommendation for improving the forensic process, and various others. The formality of the report varies depends on the investigation situation that takes place.

The log file plays a substantial role in digital forensics to reveal hidden actions of the attacker by recording its sequential steps [Chung *et al.* 2012]. It assists investigators to discover, extract valuable information, model, and analyze various events performed during the attack. In addition, investigating log files provides valuable insights through providing behavioral patterns of malicious users during their interaction with the system, network, and application. The correlation of log files is considered an important metric in investigating log files in distributed systems such as cloud computing. The correlation of log files contains various events involved in determining relationships between fragments of data, analyzing concealed data, and identifying the significance of the log files from the system, network, application and the filtered log files. Reconstruction of data from the log files and arriving towards a conclusion is also considered a part of correlation activities. As a result, log files enhance the trustworthiness and admissibility of evidence in a digital forensics process.

3. CLOUD LOG FORENSICS

Besides various log services, cloud computing provides forensic services by investigating log data to identify different vulnerabilities and malicious behaviors [Taylor *et al.* 2011]. The log data collected by CSPs are stored in a persistent, secure memory for investigating through various analytical tools and algorithms to determine different vulnerabilities inside the log files. Users can access their log data in real-time by knowing data trends and their behavior with in-depth information. To secure the log data in a cloud, a CSP uses different encryption methods to make the original log data invisible to intruders when they try to gain access [Sundareswaran *et al.* 2012]. However, CSPs have to create a level of trustworthiness to satisfy users about securing their log data in cloud computing. High performance computational resources, huge storage servers, hundreds of analytical tools, expert manpower, a fast communication network, and real-time response make users feel comfortable using cloud log-as-a-service for their log data. Sometimes, an organization knows when and where a threat has arisen, but lack of resources does not enable it to completely analyze the situation well, which then becomes costly. Today, large log-as-a-service providers ensure appropriate services for customers, including forensics, to upkeep their log data by responding with analytics, documentation, statistics, trends, charts, and graphs within user friendly GUI interfaces. According to Gartner 2015 magic quadrant for Security Information and Event Management (SIEM), Splunk and LogRhythm are considered market leaders in the data security intelligence that also provides comprehensive log management services to their clients.

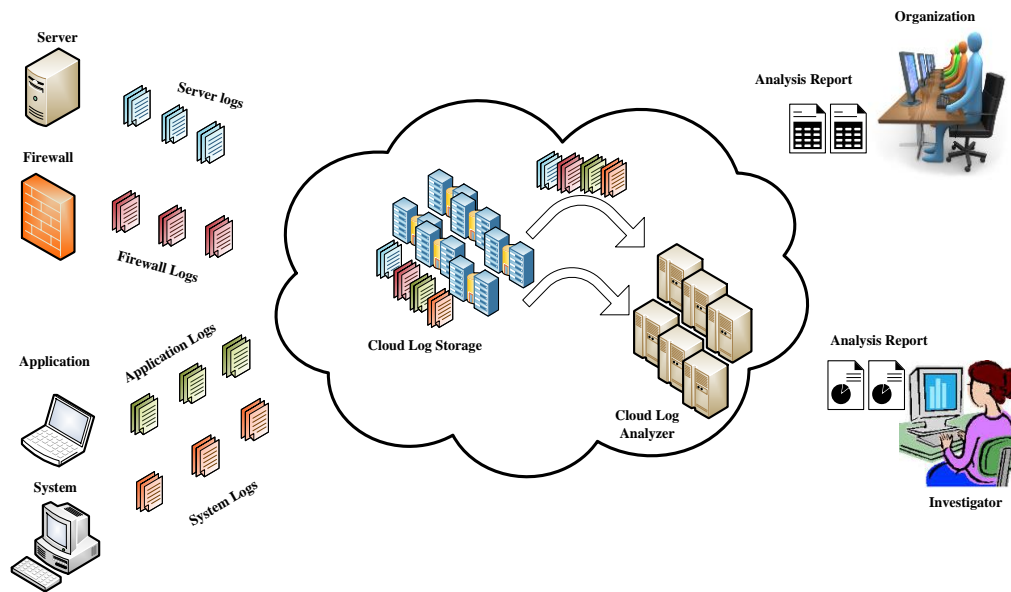


Fig. 4. Generalized cloud log forensics diagram

Cloud computing not only provides log forensic services for log files collected from outside the cloud, but also incorporates forensic services for log files collected from devices, system, operating systems, virtual machines, network, and other resources inside the cloud. For instance, the execution of an application running on an application server is logged by the CSP, or an image of a virtual machine on a resource is recorded and stored on a cloud storage resource by a virtual machine user. The generalized concept of CLF is illustrated in Figure 4. However, logging in cloud computing is not as easy as logging in to a traditional network or system [Marty 2011]. The difference comes because of the accessibility to resources. Moreover, each cloud service model has different criteria for logging depending on the data accessibility. For example, a user in an IaaS can easily collect and image virtual machine data while a user executing an application in a SaaS cannot access application logs due to restriction provided by the CSP [Sang 2013]. The application log data is collected by the CSP that is provided to the user or investigator upon the legal approval mentioned in the Service Level Agreement (SLA) between the two parties. In cloud computing, logs are mainly generated by the CSP and investigators are provided limited access to them. The dependency on the CSP makes the investigation process of identifying the root cause problems of vulnerabilities, along a specified time line more complicated for investigators. Researchers now mainly focus to minimize the dependency on CSP in investigating cloud logs data in cloud computing.

The importance of CLF is increasing due to the number of problems connected with the log investigation in the cloud [Birk 2011]. Such problems include decentralization, accessibility, storage, retention, availability, and the random log formats of the log files. The forensic investigator faces the problem of decentralization of log files due to multiple servers [Shams *et al.* 2013]. The decentralization of cloud log files creates accessibility problem such as how an investigator should access all log files stored on multiple servers at multiple locations of a single application. Log retention is also a problem for the forensic investigator in

terms of knowing how long a log file should be retained to be useful for log analysis [Popovic and Hocenski 2010]. However, log retention policy depends on CSP policies and the SLA with users, organizations, and enterprises. Similarly, the volatile nature of cloud resources (such as virtual machines assigned to users for a specific period of time) makes log files available for shorter periods of time. For instance, an application's log data disappears on the completion of an application in the PaaS service model. Moreover, log files generated at different places and having different log formats make the investigation process complicated for the investigator in terms of analyzing the log data efficiently [Ruan *et al.* 2011]. Consequently, researches have started working on the aforementioned problems, but no one has come up with a comprehensive solution or standard until today.

3.1 Cloud Log Forensics: State-of-the-art

In this section, we have classified state-of-the-art cloud log forensics into three main groups such as investigation, synchronization, and security. Each group is further compared with different characteristics that include objective, method, solution, setup, tools, and target logs. The objective characteristics highlights the main goal of the proposed solution; method characteristics explain the approach used in the solution, solution characteristics direct towards an outcome, setup characteristics describes infrastructure used to test the proposed solution, tools characteristics points application and package used in the experiment, and target logs characteristics indicates types of logs used for the experiments. Based upon aforementioned characteristics various CLF research literatures are listed in Table 6.

3.1.1 Investigation

The investigation is the primary aim of the CLF to find vulnerabilities present in cloud log files. Vulnerabilities present in cloud log files due to inadequate log management or have been generated by malicious cloud users to further exploit log files for different attacks. Currently, various researches have been conducted to investigate cloud log files.

In [Marty 2011], a logging framework is proposed to make sure the significant information generated and collected for investigators in conducting log forensic. Ensuring significant information makes the investigation process quick and efficient. In [Thorpe *et al.* 2013a], CLF service oriented architecture (SOA) framework is proposed to reconstruct various events occur in VM hosts, cloud platforms, and applications. The reconstruction of events assists the security team to identify malicious activities performed by the attacker during its attack. In [Patrascu and Patriciu 2014], a cloud logging forensics architecture is proposed to monitoring user activities in cloud computing. The layer-based architecture is used to watch each event in different layer by dividing monitoring responsibilities among the layers which helps to traceback malicious behavior easily during the investigation process. In [Ryan *et al.* 2011a], a distributed file-centric Physical Machine (PM) and VM-based logger (Flogger) is proposed to monitor the file operations in cloud computing. The Flogger collects logs from the PM and VM to deliver insight about the files accessed in the cloud. The comprehensive log information provided by the Flogger helps to identify the provenance of the files used by malicious users through analyzing events in log files.

3.1.2 Synchronization

Synchronization of cloud log files offers consistency in the log data placed at different locations in cloud computing. The consistency of log data in different log

files assists forensic investigators to check the modifications made by the attacker during the attack. Inconsistent log files may give biased results in the investigation and will not lead towards the real source of the attack. Various researches have been conducted on synchronizing cloud log files to offer reliable platform for CLF.

In [Thorpe *et al.* 2011c], transformation mapping using formal mathematical definition provides for VM log synchronization in resulting data quality assurance and invariably security. In [Thorpe *et al.* 2011d], software-based global virtual machine log auditor is developed to synchronize virtual server logs in distinct time zones in non-VM environment. The auditor used point-based and interval-based temporal data models to discuss synchronization in log files that aid investigation for malicious log files and data migration in distinct time zones of cloud computing. In [Thorpe *et al.* 2012b], formal parameterization context is used in a VM cloud environment to help forensic investigator in using synchronized VM logs as a source of credible evidence against malicious act. Synchronization of event composition in VM logs from different cloud sources is performed through binary operators such as disjunction, conjunction, and sequence. As a result, composite events of different VM logs provide enough information in identifying real sources of the attack.

3.1.3 Security

Malicious users are more interested to temper the data in cloud log files because of recorded events which may traceback the origin of the attacks. Securing cloud log files from the malicious users is a drastic challenge. The multiple and heterogeneous resources, distributed infrastructures, virtual networks, decentralized controls, and massive amount of data in cloud computing makes more difficult to secure cloud log files. However, researchers have been motivated to think out for such a significant problem which has to be addressed otherwise, will create hurdles for CLF in identifying real sources of the attack.

In [Sato and Yamauchi 2013], VM logs files are transferred in a secure way from one VM to another VM by modifying the library *'libc'* in Linux and FreeBSD operating systems. Usually, VM log file is collected by VM introspection which is not optimized for log protection. Once the VM asks for the log file is received to Virtual Machine Monitor (VMM), it takes out the logs from the kernel space and send to the SYSLOG daemon. The kernel-level malware attacks cannot temper the log files in the SYSLOG daemon. Therefore, the proposed solution assists CLF to investigate VM logs in a secured and trusted place. In [Shams *et al.* 2013], a secure logging-as-a-service is provided to the forensic investigators while ensuring confidentiality and integrity of the VM logs. The integrity of VM logs is kept by using Proof of Past Log (PPL) and the Log Chain (LC). The PPL provides temper evident scheme to prevent malicious use of altering the log files while LC keeps up the verification of correct sequence for the cloud log files offer by the CSP. The forensic investigator is benefited by having preserved cloud log files for CLF to investigate malicious behaviors. In [Prabha *et al.* 2014], homomorphic encryption scheme is used to encrypt the cloud log files to hide data from malicious users. However, cloud operation can be performed on encrypted log data without exploiting confidentiality and privacy of the legitimate user data [Khan *et al.* 2015]. The forensic investigators are confirmed about originality of the log files because log files are encrypted before sending towards them. It helps in identifying the real source of the attacks through analyzing different logs get from the cloud upon the detection of malicious events. In [Shams *et al.* 2014], Forensics Aware Language (FAL), a domain specific language is developed which is applied to secure logging of any log

Table 6: Classification of cloud log forensics

Classification	Objective	Method	Solution	Setup	Tools	Target logs	References
Investigation	To provide a proactive approach to ensure the generation of logging for forensic investigation	Implementation of application logs in SaaS	Three-tiered setup on top of cloud infrastructure	Testbed	Django, JavaScript, Apache, MySQL	Application logs	[Marty 2011]
	Investigation of cloud logs based on forensic-based service oriented architecture.	Cloud actor interaction scenario	Cloud audit forensic framework	Theoretical Explanation	N/A	Various logs	[Thorpe <i>et al.</i> 2013a]
	Solution to assist investigators to monitor user activities in cloud computing	Cloud logging architecture	Layer based forensic	Testbed	Apache HTTP server, PostgreSQL	VM logs	[Patrascu and Patriciu 2014]
	To monitoring file access and transfers within cloud computing through centralize logger	Flogger: A File-centric logger	logging file life-cycle on both VMs and PMs	Testbed	PostgreSQL 9.0, MonetDB	VM logs, physical machine logs	[Ryan <i>et al.</i> 2011a]
Synchronization	To establish VM log synchronization	Formal constraints	Transformation mapping	VMWare esx3i Data Center	N/A	VM logs	[Thorpe <i>et al.</i> 2011c]
	Synchronization of VM logs in different time zones such as non-native VM environment	Formal temporal mechanism	Global VM log auditor	VMWare esx3i Data Center	N/A	VM logs	[Thorpe <i>et al.</i> 2011d]
	Synchronization of log events in distributed forensic computes cloud database environment	Formal parameterization	Synchronized log event composition	VMWare esx3i Data Center	Global virtual machine log auditor	Hypervisor System logs	[Thorpe <i>et al.</i> 2012b]
Security	Securely transfer logs from one VM to another VM to protect from tampering	Replacing library in the VM	Writing additional code to <i>libc</i> library	Testbed	N/A	VM logs	[Sato and Yamauchi 2013]
	Provide confidentiality and privacy of the cloud user data	Schematic description	Past log proof generation and verification	Prototype	OpenStack, Snort	VM logs, Network access log	[Shams <i>et al.</i> 2013]
	Execute queries on cloud logs without effecting confidentiality and privacy	Homomorphic encryption scheme	Anonymous tag generation	Prototype	Log generator: Self developed	Various logs	[Prabha <i>et al.</i> 2014]
	To apply secure logging mechanism on any logging mechanism	Forensics Aware Language (FAL)	System & application logging	Programming development	Development of FAL compiler using LISA	Various logs	[Shams <i>et al.</i> 2014]

format. FAL used hashing to get integrity of the log files. The integrity of cloud log files facilitates CLF to get correct results through analyzing original log files. Moreover, using FAL, own log structure can be defined and is parsed to the log file based on the defined log structure. This feature helps forensic investigators to overcome the problem of heterogeneity of log formats faced during their investigation process.

4. LOG-AS-A-SERVICE: CLOUD LOG MANAGEMENT

Logs are records for capturing various events occurring in a system, network, or process along a specified timeline [Chuvakin *et al.* 2013]. Each record in the log specifies information related to the sequential steps occurring during the time of system, network or process execution. The increase in various logs makes organizations adopt log management for the appropriate handling of logs within the existing infrastructure. However, the increased size, number, and frequency of logs make difficult for an organization to manage logs within the context of scarce resources, administrative staff, and security approaches.

The best option to cope with the aforementioned problems is to use the “log-as-a-service” services of cloud computing [Abadi 2014]. Nowadays, many organizations use the log services of a CSP to simplify their log management. The CSP log-as-a-service assists organizations in managing logs, such as integration of operational log data from various locations, instant log visibility, monitoring of logs in real-time, search and filter log data, and much more. Organizations use log-as-a-service services by simply passing different logs to a CSP for managing inside the cloud infrastructure. The log files are transferred to the cloud in different ways depending on log management of the CSP. For instance, logentries provides customer with multiple options to send their log data to the cloud server i.e. agent-based logging, SYSLOG forwarding, application-based logging, and token-based logging. The agent-based logging contains lightweight agents installed on client side provided by logentries to automatically collect and send log files to the cloud servers. SYSLOG forwarding used operating system logs forwarder to send log files to the cloud servers. Application-based logging is performed through in-application logging provided to collect logs using different programming languages. Token-based logging provides to integrate multiple log instances from different places into a single container in the logentries user interface. This method is used for large organizations which have to log data from different distributed locations. The CSP provides different log analyses for the organization while using high computational resources, high potential analytical tools, and cloud resources. The CSP uses high computational resources by combining thousands of computers in different data centers. For instance, Amazon used 26,496 CPU cores, 106TB of memory, and a 10 Gigabit Ethernet interconnect to build a high computational cluster. Similarly, high potential analytical tools such as sumo logic, event tracker, Scalyr, and others are used by CSP to perform in-depth log analysis in providing useful information to their customers. The log-as-a-service saves the time, cost, and experts required by an organization to analyze the log data. The subsequent section explains some of the CSPs that provide log-as-a-service to users and organizations from different perspectives. A brief description about the comparison of CSP log-as-a-service is described in Table 8.

The comparison of CSPs providing log-as-a-service has been done according to various parameters which highlight the core competency of each. The comparison parameters include forensic, access, price, mobile platform, custom logging, crash

logging, dashboard, log format, encryption, security, advantages, capacity, OS support, and installation which are briefly described in Table 7. The forensic parameter indicates the investigation facility provided by CSPs to their users in terms of log records. As shown in Table 7, the CSPs provide forensic investigation for the detection of any intrusion and vulnerability found in the various log records. The access parameter indicates whether the log-as-a-service is an open source or whether it is provided under a proprietary trademark. The price parameter helps users to know whether the log-as-a-service provided by the CSP is paid for or free (free trial). The mobile platform parameter shows the mobile operating systems supported by the various CSPs for their log-as-a-service such as iOS or Android. The custom logging parameter indicates that users can decide what should be included in the log file to fulfill their requirements [Samudra 2005]. Therefore, different users can have different log fields in their log files. Similarly, the crash logging parameter specifies the logging facility which captures the current state of the system before the system goes down (crashes) [Yang *et al.* 2014]. Crash logging is very useful in a situation where the system frequently crashes. The dashboard parameter shows the GUI provided by the CSP to view log data analysis in an easy and simple way. Log format parameter indicates what types of log format access are allowed by the CSP to log data. For instance, does it provide a single log format or customized log format according to users' requirements, where users can build their own log format. The encryption parameter indicates the encryption algorithms applied to log data to protect it from different attackers. Similarly, the security parameter shows the secure access provided by the CSP to users' log data in the cloud. The advantages parameter indicates the core competency services of the CSP in providing log management services to users. The capacity parameter highlights the volume of log data managed by the CSP. The OS support parameter indicates the operating systems supported by the CSP for the log-as-a-service. The installation parameter shows the level of difficulty in installing and configuring the CSP log-as-a-service client.

Table 7: Description of the parameters used to compare the Log-as-a-services solutions

Comparison	Description
Forensic	Investigation facilities provided by CSP to analyze log files for various vulnerabilities.
Access	Users freely contribute to the log-as-a-services through accessing open source codes or it is restricted by CSPs to having commercialized licenses.
Price	Indicates either the log-as-a-services are freely provided by CSPs to their clients or they charge an amount to provide the logging services.
Mobile Platform	Log-as-a-services provided by CSPs is accessible on mobile devices using mobile applications.
Custom logging	A facility provided by CSPs for its users to modify log files content based on their requirements.
Crash logging	Services provided by CSPs to restore log files from its previous saved state upon crashes of log files.
Dashboard	The GUI provided by CSPs to facilitate users in accessing log analytics through graphs, charts, and statistical results.
Log format	The CSP provides single or multiple log formats to make log files.
Encryption	The log data is secured in log files.
Security	Secure channel provided for users by CSPs to access log files in the cloud.
Advantages	The core benefits provided by CSPs to users in terms of log services.

Comparison	Description
Capacity	The volume limit provided by the CSP to log the data in log files.
OS support	An operating system used by CSPs in providing log-as-a-services to different users.
Installation	Level of efforts is required by the users to configure log-as-a-services acquired from the CSP.

4.1 IBM SmartCloud Analytics

IBM SmartCloud analytics are a log analysis framework that uses the IBM cloud infrastructure to analyze the operational data of an enterprise integrated with various sources [Ellis 2013]. It helps in identifying, isolating, analyzing, and resolving operational data related issues associated with logs, support documents, events, and metrics. Moreover, it reduces the processing time needed to perform root cause analysis by implementing quick search, filter, and visualization of the data in a single application interface. Various logs, including Web logs, Windows logs, Syslogs, and Delimiter Separated Value (DSV) logs, are integrated with significant log services to perform accurate and quick log analysis. For instance, Logstash, an open source log management integrates with different type of logs collected at different locations provide centralized processing of log data, normalize various data and schemas, extended customize log formats, and add a plugin for custom data sources [Sissel 2014]. This result in an accurate and quick log analysis of the log files collected from distributed locations. IBM SmartCloud analytics-log analysis incorporates more features that make it one of the premier log-as-a-service providers in the market, with improved service availability, decreased mean time for repair, dynamic warning messages, separation of issues related to specific domains, rapid index search, and visualized search results.

4.2 Papertrail

Papertrail provides log-as-a-service to users via browsers, API, and the command line interface [Lindvall 2014]. Papertrail's main objective is to provide hosted log management for various log data integrated from different sources, including SYSLOG, text log files, apache, MySQL, windows events, routers, and firewalls. The text log files are treated by Papertrail using file systems which are inaccessible via command line, web or email. The required data in the text log files are isolated and distributed on multiple applications, systems, and directories for instant processing and security purposes. Papertrail ensures the security of log data by providing TLS encryption and certification-based verification for the destination host. At the end of each day, Papertrail automatically archives log messages and metadata to Amazon S3 and provides an optional choice for users to store one copy in the bucket that is provided. A user has full access to view the log record in the provided bucket, which is controlled by AWS. The logs created by Papertrail are in Gzip compressed format with tab separated values, for example:-"Tape/Papertrail/logs/98765/dt=2014-12-24/2014-12-24.tsv.gz". The 'Tape' is the bucket name, '98765' is the Log ID, 'dt=2014-12-24' is the date, where '2014-12-24.tsv.gz' is the Gzip compressed file extension with the specified date. Moreover, Papertrail integrates with other services to enhance log management services for their users i.e. Amazon Simple Notification Service [Amazon 2015], Boundary [Heath 2014], GeckoBoard [Simon 2014], OpsGenie [Mollamustafaoglu 2014], Slack [Butterfield 2014], and others.

4.3 Logentries

Logentries is a cloud-based company from Ireland, which provides software services for log management and analysis based on different user demands [Burton 2014]. The main objective of Logentries is to deliver real-time log analysis outcomes with fewer time delays and greater user satisfaction. Logentries collect different logs and analyzes them through software stacks while using pre-processing steps such as filtration, correlation, and visualization of log data. The intuitive log search of Logentries assists the user through the writing of simple keywords, regular expressions, and phrases. Logentries provide an anomaly detection facility to determine the changes occurring within the parameters of the search queries from time to time. The multiline graph services of Logentries help users to create a single view for multiple search queries. They assist users, forensic investigators, enterprise owners to view many search query outcomes in a single interface with organized and structured data. Moreover, Logentries incorporates other framework features to further help the user through delivering well developed services i.e. Django [Holovaty 2014], Grails [Rocher 2005], node.js [Dahl 2014], Sinatra [Mizerany 2014], and Heroku [Nielsen 2014].

4.4 SplunkStorm

SplunkStorm is a cloud-based log management software that helps users in monitoring, diagnosing, and troubleshooting various cloud applications, executing on different platforms including AWS, Google App Engine, Heroku, Rackspace and others [Baum 2014]. SplunkStorm gathers machine data generated by servers, websites, applications, as well as click stream data, call records, web transactions, and various network activities. The collected data is sorted to identify and resolve different kinds of application issues. SplunkStorm services help users to perform searches on historical as well as current machine data, filter specific events, link transactions of different application components, correlate data of different data types, determine the trend analysis of various operational parameters, share their own projects with friends and colleagues, and generate reports of data for resolving inside data issues. SplunkStorm is best utilized by developers in terms of generating statistical analysis for applications, analyzing various events through semantic logging, search and squeeze application, and performance bugs. The semantic logging is the method used to create consistent log structures using strongly typed events. The semantic logging makes easy to query and analyze log data due to its reliable consistent format and structure. Similarly, SplunkStorm also assists in monitoring application availability and performance, monitoring user activities, and identifying risk patterns for various threats such as data leakages and brute force attacks.

4.5 Loggly

Loggly is a US-based cloud log management service provider which aims to provide easy access with centralized analysis of the log data to their customers [Oppenheimer 2009]. Loggly collects log data directly from various sources or devices, i.e. routers, firewalls, servers, storage devices, different hosts and generates a visualize reports in real-time. Loggly help users to check the status of their applications, websites, and services, and how they act according to different time bases. In 2013, Loggly launched its 'Generation 2' services to provide new analytical tools, interfaces, point-and-click graphs, advanced searches, automated event parsing, and scaled out architecture to efficiently manage users' data. It is not an easy job for a company to collect and analyze millions of events on a daily basis, which might

require huge infrastructure. Loggly even assists customers to view trend analyses of their log data for searching various issues and events by accessing the visualized interface via the web browser. The easy and simple log management services make Loggly a more attractive option among the various cloud-centric application companies. As a result, at the end of the year 2014, Loggly had logged more than 750 billion events, processed more than 250 TB log files, and had more than 21 thousand active accounts. The incorporation of value added services by Loggly attracts customers to use the services for their cloud-based applications to log their data for better operational performance and to determine security related issues such as threats and risks.

5. USE CASE STUDIES OF A CLOUD LOG FORENSICS

Case studies are considered a research strategy to investigate a tool, project, process, system, services, etc. empirically to determine the effect in a real life situation [Gerring 2007]. Here, in this selection we explain five case studies related to CLF provided by various CSPs providing facilities for investigating different logs for vulnerabilities. Table 9 highlights the main features of each CSP mentioned in the case studies in terms of delivering CLF.

Each case study is compared with different characteristics such as: a) case study type b) focus, c) cloud technology, d) log type, e) advantage, and f) outcome. The case study type characteristics show the nature of the case study i.e. company-oriented. In our case, the focus characteristics contain various objective values of different case studies that include copying HTTP logs into Amazon S3, identify the root cause of attacks, backing log data, identify suspicious content, and investigation of malware in web pages. The cloud technology characteristics contain various platform values used in case studies that include HP ArcSight Logger, Dynamic Field Explorer, Elastic Map Reduce (EMR), Amazon S3, and Rackspace.

The HP Arcsight Logger is a log management tool used to collect, store, and analyze machine data from any device, source, and vendor platform. Its build-in rules and report enables monitoring, detection, alerting, and forensic investigation for security measures. The Dynamic Field Explorer (DFE) is a new approach to the log analysis provided by the Loggly to differentiate between the most common events and anomalies in the log files. Its comprehensive summary, in-depth log analysis, easier and faster management, and statistical report help investigators find the root cause of the problem easily. Amazon EMR is web services used to provide processing and analysis for the huge amount of data. It used MapReduce framework to process data parallel in a distributed environment. The Amazon EMR is used for different data analysis that includes log analysis, financial analysis, Bioinformatics, and various others. Amazon S3 is high volume object-based storage system provided to the users through web in secure, scalable, and durable manner. The user can store and retrieve data from anywhere through a simple interface on the web by paying only for the storage devices used. Rackspace is the CSP with an aim to manage everything what they provide. Rackspace provides a multi-tenancy platform to different users having different requirements, having almost 100% network uptime, and manage redundancy based on users need.

The log type characteristics contains different targeted log values used in case studies for investigation that include web logs, system and application logs, HTTP server logs, net flow logs, and access logs. The advantage characteristics contain values of extra features obtain using respective approach that includes scalability, robustness, fault tolerance, flexibility, cost-efficient, and opportunity cost. The

“scalability” value indicates the current technology used in case studies can be extended for large amount of log files. The “robustness” value indicates the current system can cope up with malicious state occur during an investigation of log files. The “fault tolerance” value indicates the system provided for the investigation of log files can work at the time of its failure. The “flexibility” value indicates the integration of different technologies used with the current log investigation system. The “cost-efficient” value indicates reduction of the operating cost for a log investigation system. The “opportunity cost” value indicates to avail alternative benefits with less cost. Furthermore, outcome characteristics have two values, such as: a) success and b) moderate. The “success” value indicates that the case study was successfully implemented with achieving its objective while a value “moderate” indicates that the case study was implemented without achieving completely its objectives.

5.1 Heartland Payment Systems

The Heartland Payment Systems (HPS) is one of the fifth largest payment processor companies in the US to process more than 11 million transactions per day, with a monetary value of around \$80 billion per year [South 2013]. Beside payment processing, HPS provides other multiple services such as payroll, e-commerce, mobile ordering, school payments, lending, etc. in different industries including restaurant, hospitality, petroleum, retail, and education. Based on the multiple financial services of the business, HPS was constantly exploited through various vulnerability probing attacks. It was a great challenge for the HPS to investigate vulnerabilities in an enormous amount of log data collected during financial transactions. In 2009, HPS was targeted with a SQL injection attack that stole 130 million credit and debit card numbers of different users from network and computing resources. The HPS was fine \$60 million by Visa Corporation and its operation was suspended for six weeks, which put them down by losing many loyal customers. As a result, huge financial penalty and customer loss, HPS decided to strengthen its security by focusing to analyze activities on network and computing infrastructure to find the root cause of the malicious patterns at the early stage of its occurrence. HPS acquire the HP ArcSight logger services from the Hewlett-Packard to gain insight of potential threats across its infrastructure by analyzing their log files. HP ArcSight logger incorporates HP Cloud Service Automation (CSA) to provide log forensics services using the cloud infrastructure. HP ArcSight logger offers an ultra-fast log forensics service that unifies full-text searching, alerting, analysis, and reporting across entire enterprise machine data provided in the log files.

In addition, the Security Information and Event Management (SIEM) provided by HP ArcSight logger has expedite log forensics by reducing the time-frame to respond malicious activities quickly and limit the manpower cost by focusing on the source of the alert rather than utilizing multiple teams to mobilize to investigate suspicious events. Using HP ArcSight logger, HPS investigators have determined different security threats by analyzing various logs of the infrastructure in a real-time, which have been overwhelmed before it affect the victim. The HPS investigators gain benefits by using HP ArcSight logger to have log data collection from a numerous set of sources, ease deployment of log forensics, ultra-fast forensics through full-text searching, ongoing monitoring, flexible log storage options through highly compression ratio i.e. 10:1, and real-time analysis of large number of log files. Therefore, HPS has protected and grow its business significantly by using HP ArcSight logger and wins many industry awards, including John South, Chief Security Officer (CSO) of the year, 2013 by SC magazine.

5.2 Monex Financial Service Provider

The Monex Company is an online financial services provider, based in Tokyo, Japan with having several online securities trading subsidiaries. Monex provides financial trading services to more than 1.5 million customers in Japan [Beaver 2015]. The web application used for financial services has been developed in windows development stack with .Net front end application and MySQL database. Monex depends on the application log data to identify malicious behavior of the attack at the time things are not running as expected. The challenge faced by Monex was to analyze huge amounts of log data in a real-time to determine the root cause of the attack. However, Monex fails to achieve efficient and fast investigation mechanism to cope up with huge amount of log data in a real-time.

Monex start using DFE, a service provided by the Loggly, a cloud log management provider. The DFE provides a complete structural summary of your log data which helps to differentiate between common events and anomalies, as well as provide quick and precise way to filter into specific logs. The Monex security investigators benefits from DFE to perform automated log parsing, in-depth log analysis, sanity checks, and root cause identification. Moreover, the DFE real-time event count feature aids Monex security investigators to understand the magnitude of the problem faster and determine the location where the problem exists. This leads towards quick and efficient threat response to the correct part of the system. Mostly, the faster responses are performed on the occurrence of MySQL connection errors, connectivity issues with back-end data sources, and application errors based on various attacks. As a result, Monex gain relevant insights into the log data to investigate malicious events present in the system much accurate and faster. The quickly pinpoint of malicious events in log data through DFE makes Monex more productive by focusing on its core competencies rather than analyzing log for malicious sources.

5.3 Banca Intesa Bank

The Banca Intesa is a leading bank in Serbia having more than 1.7 million clients. The bank processed more than 11 million events per day approximately collected from various log files from different network equipment, security devices, and databases [Stanojevic 2013]. As a financial institute, Banca Intesa believes to be secure from vulnerabilities exploiting different customer records through various attacks. The Banca Intesa was looking for centralized log management services to provide complete analysis of user and network activities. The aim of the Banca Intesa was to correlate information collected from different part of the bank infrastructure and perform investigation through root-cause analysis resulting in response to suspicious events and potential threats.

Banca Intesa acquires the services of HP ArcSight logger to search collected logs for potential threats that cause damage to the bank infrastructure. HP ArcSight logger delivers a comprehensive log forensics services to the Banca Intesa bank through analyzing various log files, including critical events in real-time, alert notifications, correlation of log information, data and user monitoring, application monitoring, and threat intelligence. The log forensics information helps security investigators of Banca Intesa bank to know who is on the network, what data have been accessed, and what actions is performed on the data. The information obtained from the log forensics helps security investigators of the bank to control the malicious actions performed by the malicious users before it continue to damage the bank infrastructure. The HP ArcSight logger provides strong security foundation for the

Banca Intesa bank to analyze their log files in finding the root cause of the threats in a real-time.

5.4 Yelp Content Analytics System

Yelp is a corporation with a website that publishes crowd sourced reviews about local businesses [Stoppelman 2004]. In the first decade, Yelp spread its business to 29 countries with more than 120 markets. Yelp has more than 130 million monthly users who visited the website for different purposes, including business reviews, updating business contact information, generation of business plans, updating of personal life experiences, and much more. To log such a massive amount of data is difficult and further requires data analysis to determine malicious behavior. Yelp started using Amazon S3 and Amazon Elastic Map Reduce (Amazon EMR) to overcome the aforementioned problems. Yelp reported that 1.2 TB log data is stored on Amazon S3 on a daily basis. Yelp uses Amazon EMR to process the log data for analyzing suspicious content. Amazon EC2 assists Yelp in performing log analysis to determine suspicious content, and therefore saves innocent users. As a result, Yelp saves upfront hardware costs by utilizing Amazon EMR for analyzing log data and moreover, focuses on opportunity costs to gain more with less cost with the concept of pay only for what you use.

5.5 Buzz Coffee

This case study relates to a malicious user that exploits a victim by generating a malicious webpage on the cloud [Dykstra and Sherman 2011]. The hacker uses a Buzz Coffee company website to generate the malicious payload by installing a rootkit. By doing so, he hides himself from being detected by an operating system. Moreover, users are directed to the malicious content of the website from where it performs a malware attack on them. To investigate such malicious attacks is a challenging task. The investigators generate a court order to investigate the logs of the cloud. The CSP provides access logs, Net flow logs, and a web-server virtual machine file at the request of investigators and refuses to provide raw data citing proprietary and confidential information. The integrity of files and logs are verified through performing a hash algorithm. The data collected from the CSP are compared with the original data on the Buzz Coffee website to identify the differences between them. The second option is to create a timeline for the whole process. Web access and Net flow logs combined together provide useful metadata regarding malicious users, i.e. login time, number of access, IP address and others. However, the lack of raw data raised several questions in the court against CLF process conducted for Buzz Coffee website, such as: (a) Is the chain of custody achieved? (b) Does the IP address identified really belong to the hacker? (c) Does the CSP provide complete log data? and (d) Which mechanisms are used by the CSP to secure the infrastructure? The CSP does not provide raw data related to the operating system information that creates ambiguity in the investigation process

Table 8: Comparison of different cloud log service providers

Comparison	IBM SmartCloud Analytics	Papertrail	Logentries	Splunk Storm	Loggly
Forensic	Yes	Yes	Yes	Yes	Yes
Access	Proprietary	Proprietary	Proprietary	Proprietary	Proprietary
Price	Paid, 90 days free trial	Paid, 60 days free trial	Paid, 30 days free trial	Paid	Paid, 30 days free trails
Mobile Platform	n/a	iOS, Android	Android	iOS	iOS, Android
Custom logging	n/a	Yes	Yes	Yes	Yes
Crash logging	n/a	n/a	Yes	Yes	Yes
Dashboard	Yes	Yes	Yes	Yes	Yes
Log format	Customize	Customize	Customize	Customize	Customize
Encryption	Advanced Encryption Standard (AES)	TLS encryption	Diffie–Hellman key exchange	Advanced Encryption Standard (AES)	TLS encryption
Security	SSH Key-based authentication	Certificate-based verification	Secure Socket Layer	Third party solution (Meldium, Bitium)	HTTP/S using RESTful API
Advantages	Root cause analysis, Isolate issues	Instant alerts, long term archives	Anomaly Detection, Multiline graphs, shareable dashboard	Availability, Data privacy and security	Easy logging without installing agent, streamline log analysis
Capacity	Unlimited	500 GB	Unlimited	20 GB	Unlimited
OS support	Red Hat Enterprise Linux Server	Unix/Linux	Windows, Linux, Mac	Windows, Linux	Windows, Linux, Mac
Installation	Medium	Easy	Easy	Medium	Medium

Table 9: Summary of different cloud log forensics case studies

S. No	Case Study	Case study type	Focus	Cloud Technology	Log type	Advantage	Outcome	Reference
1	Heartland	Company-oriented	To investigate malicious activities across entire infrastructure and overwhelmed them before they do damage	HP ArcSight Logger	Network logs, Server logs	Scalability, reduce business risk	Success	[South 2013]
2	Monex	Company-oriented	To analyze huge amounts of log data in a real-time to determine the root cause of the attack	Dynamic Field Explorer	Application logs	Quick response, Improve usability	Success	[Beaver 2015]
3	Banca Intesa	Company-oriented	To investigate root cause of the attack resulting in real-time response to suspicious events and potential threats.	HP ArcSight Logger	Network logs, Security logs, Database logs	Comprehensive user activity monitoring	Success	[Stanojevic 2013]
4	Yelp	Company-oriented	To identify suspicious content	Amazon EMR, Amazon S3	Web logs	Scalability, opportunity cost	Success	[Stoppelman 2004]
5	Malicious webpage	Company-oriented	To investigate malware in web pages inside the cloud	Rackspace	Net flow logs, Access logs	Scalability	Moderate	[Dykstra and Sherman 2011]

regarding the identification of the malicious user generating malware. Therefore, the case was hindered and Buzz Coffee fails to determine the real hacker of the website attack.

In real life investigation situations, CLF has to ensure complete access to the cloud data available on the cloud resources. The investigators should have good relation with CSP to guarantee in providing complete data from the cloud resources. However, it does not happen in cloud computing because CSP has different level of SLA with a different users, SOP, privacy levels, and trustworthiness. Based on the parameters, CSP is bound not to provide information to the investigators. Therefore, the best option is to provide forensic-as-a-services by the CSP to different users/companies depending on their requirements.

6. CLOUD LOG FORENSICS: SECURITY REQUIREMENTS, VULNERABILITY POINTS, AND CHALLENGES

As per the discussion in Section 2.2, CLF is becoming a significant process for the security of cloud computing. Massive log generation at multiple locations increases the demand for storage space in an organization. An organization with scarce resources is not able to accommodate a massive amount of logs, which encourages it to migrate towards cloud storage [Shiraz *et al.* 2015]. However, storing important data in third party resources in cloud computing creates a risk for an organization in terms of data protection. Publicly available cloud computing adds more risk to data protection in terms of the easy and simple accessibility of cloud resources [Ramgovind *et al.* 2010].

An attacker can hire multiple resources in the cloud and use resources for attack generation by accessing log databases to delete and alter log data files. To minimize threats from the exploitation of cloud logs in cloud computing, the CSP has to protect users' log files and has to provide comprehensive and adequate CLF. The forensic process of investigating log files in cloud computing will help the CSP prevent attacks in the future. Similarly, a generated forensic report at the end of the investigation process is sent to the organization. The level of trust is increased between the CSP and the organization in terms of performing adequate forensic processes for cloud log files.

In this section, the CLF is classified into three parts including security requirements, vulnerability points, and challenges. CLF requirements show the security parameters necessary for the cloud log to be investigated for valid (original) evidence. CLF vulnerability points include places where attacks can take place to exploit cloud log files inside or outside the cloud. In the last part of this section, CLF challenges are surveyed. Although some solutions have been proposed for a few of the challenges, due to the small amount of consideration given to such challenges, more research efforts are required to ensure adequate and practical outcomes.

6.1 Cloud Log Forensics: Security Requirements

It is unacceptable to provide a secure CLF environment without considering security requirements for the cloud logs. The requirements include confidentiality, integrity, availability, authenticity, and privacy. Each CLF security requirement is described in subsequent paragraphs and highlighted with a description in Table 10.

Confidentiality. Confidentiality deals with the preservation of user data in the cloud log files. Sensitive data in the cloud log files should not be disclosed to any

individual. The individual could be an attacker or another CSP. In analyzing cloud log files, there may be sensitive data available about the user, including password, credit card number, content of emails, and others. Such sensitive information creates security concerns for a person who investigates cloud logs and similarly for a person who accesses cloud logs legally or illegally. Likewise, in CLF, confidentiality is also exploited when one log file contains many users' data at the same time. Retrieving one user's data provides the opportunity to access other user's data in parallel either intentionally or unintentionally. As a result, when carrying out CLF, the CSP should ensure that user data is protected from any sort of violation that would destroy the level of trustworthiness including violation of user confidentiality.

Integrity. The integrity is considered a vital parameter for cloud log files in relation to providing evidence against attackers. Integrity deals with the non-tampering or non-modification of cloud log files after they are generated and stored in the cloud [Yun *et al.* 2014]. Improper secure cloud log storage and transit might create susceptibility to destruction and alteration of cloud log integrity. As a result, variety problems are created, including unnoticed malicious activities, manipulating of evidence, hiding of malicious users and others. For instance, there are specifically designed rootkits, which alter log file data to modify rootkit execution and installation. As a result, during the CLF process, an investigator or CSP should provide evidence to the court after the investigation that the whole process was conducted based on original cloud log files rather than tampered ones.

Availability. Availability deals with cloud log data that must be available whenever required [Yin 2014]. In cloud computing, cloud log files are replicated to more than one place for the sake of security and reliability. However, the availability problem arises when the attacker has access to a cloud log file before it is replicated to various other resources. The accessibility of cloud log files to attackers might result in the deletion of log files to hide their identity. Similarly, availability is also affected by the log data retention policies of an organization. For instance, a log has a specified maximum limit which indicates the volume of the log data. The limit should be in capacities such as 500 megabytes or it can be in numbers such as 100,000 events. Once the limit is reached, the logs are overwritten or logging stops, which causes loss of data. Therefore, it results in minimizing the availability of cloud log files. Consequently, CLF availability is essential to investigate log files with complete and accurate data.

Authenticity. Authenticity deals with accessibility permission to cloud log files. The CSP has to ensure the cloud log files are only accessed by authorized individuals having justifiable objectives such as investigation. Sometimes, a cloud log file is accessed by an investigator or CSP employee; however, she may delete or alter some part of the log file affecting the entire process of CLF. The CSP has to verify with the court that the cloud log files are accessed by individuals having legal permission or have been assigned by a third party investigation agency. Similarly, the right person has to access the right cloud log file while searching the massive amounts of log files in the database in cloud computing. Accessibility to non-authorized cloud log files would leak other users' information that would reduce the trust of users in accessing their data. Complete accessibility to cloud log files should be maintained in the form of a report by the CSP recording each and every access to log files stored in cloud computing.

Privacy. Privacy deals with securing user log data at every stage of CLF from the generator to the analysis stage. In cloud computing, each physical resource has multiple virtual machines which have multiple user applications running at the same time, such phenomena are known as multi-tenancy in cloud computing [Jahdali *et al.* 2014]. Logs generated in a multi-tenant environment contain many users' data at the same time. The multi-tenancy environment of cloud computing makes the investigation difficult for investigators to isolate data from various resources at the time of investigation [Simou *et al.* 2014]. The probability increases to access an innocent user's log data while accessing malicious users' log data files. Ethically, an investigator or CSP should access the log data of the malicious user which is required for the investigation while avoiding accessing other log data due to possible violations of data privacy rules and regulations. As a result, in CLF, privacy is a key requirement and a challenge for forensic investigators to keep intact.

Table 10: Cloud log forensics: Security requirements

Cloud log forensics security requirement	Description
Confidentiality	To provide a security for cloud logs generated from various sources through preventing unauthorized access.
Integrity	To safeguard cloud logs from being altered or modified by authorized or unauthorized person performed an action intentionally or unintentionally.
Availability	To guarantee cloud logs available for analysis in the original form as it was created and stored.
Authenticity	To assure right user to have access to have authorized access towards the cloud logs store in the cloud.
Privacy	To preserve user's data from leakage during generating, collecting, storing, and analyzing cloud logs in the cloud.

6.2 Cloud Log Forensics: Vulnerability Points

CLF strongly relies on important security features for log data such as confidentiality, integrity, and availability. An investigation of log data must preserve the sensitive data of the user presented in the cloud log while analyzing it for various susceptibilities. Similarly, an investigation should resist deleting, and modifying any sort of data in the cloud log so as not to compromise the integrity of the data. However, the availability of the log data is also significant due to the need for robust log analysis with accurate and timely identification of vulnerabilities.

Different vulnerabilities are generated by attackers on cloud logs in order to perform malicious activities with the aim of destroying their attack traces, modifying and deleting log data, diverting the investigation process in other directions so as to hide them, extracting sensitive data, and others. Now, our focus in this section is to explain the possible vulnerable points in the cloud logging infrastructure. We have divided the cloud logging infrastructure into five parts to clearly highlight the entire attack process on the cloud log at different log locations. The potential vulnerability points in the cloud logging infrastructure include log generation, log collection, network, log storage, and log analysis. Each of the vulnerable points in the cloud logging infrastructure are described and illustrated in Table 11.

Log Generation. Log files are generated through various tools and configurable files, e.g. ProcMon.exe, vmware.log file, and aLogcat pre-configured to capture require information from servers, network, devices, and applications. Cloud log files are updated with log content with a passage of time when the system,

process, and network starts its execution in the cloud. In cloud computing, CSP builds log files in various locations in the cloud to record different events including virtual machines, hosts, servers, networks, and various applications, in order to record different events along a specific timeline. Each above mention entity creates a log file depending on the pre-defined log generation configuration provided by the CSP. Moreover, in the huge infrastructure of cloud computing, it is difficult to find the exact location where logs are generated. However, the accessibility of logs generated systems or applications in cloud computing to an attacker could affect the availability of cloud logs for CLF. The attacker could destroy the log generated application or system by deleting the configuration files, injecting malicious code, forcing it to perform malfunctions, misdirecting it from the objective. However, the confidentiality and integrity of the cloud log data in such situation is not an issue due to the intention of the attacker to destroy or delete the execution files of the log generation application or the system rather than looking at cloud log file content.

Log Collection. The log files are collected by the cloud log collector or cloud agent from various sources in cloud computing. After generating different cloud log files, the cloud log collector collect cloud log files to store them on different resources in cloud computing. However, once the attacker gets access to log collection locations, he/she can easily exploit cloud log files. At this point, cloud log files are available for the attacker to delete or modify by removing the attack traces while compromising confidentiality, integrity, and availability. The log collector mainly collects cloud log files in zip format, which can be easily converted by the attacker to normal log format for understandability. Third party log collectors must ensure their security strengths to avoid such kind of accidents happening with cloud log files which could create a question mark against CLF in the court.

Network. The network, also known as transit, is used to carry cloud log files from cloud log collectors to the log storage resources. The easiest way for attackers to attack is to interfere between cloud log collectors and cloud log storage resources rather than breaking the security hurdles for each. The network is a medium to connect two or more resources, systems or general points, which does not belong to any of the parties. In the case of cloud log attacks, the attacker wants to capture the data passes on the network to interpret cloud log data files in terms of their usefulness. Getting access to cloud log data files on the network could provide the sensitive data of a user, entire data recorded along a specific time line, understanding of the whole process, and others. Confidentiality is compromised due to leakage of the data, whereas integrity is compromised due to modifying and altering data on the network. Similarly, availability could be affected by deleting some or all of the cloud log files while passing from cloud log collectors to cloud log storage.

Log Storage. Log storage is the location/resource where cloud log files are stored, to be analyzed in the next stage of CLF, such as cloud log analysis. The security of cloud log files stored on cloud resources depends on the security provided to them in terms of log format, encryption, authentication access, and others. The log format used to store cloud log files in storage might be different from the log format used at log generation and log collection. The attacker might have access to some of cloud log files at cloud log collection and now he wants to have access to more information from the logs at the storage location, but he may be restricted from doing so due to the different cloud log format. Similarly, most of the log-as-a-service

provider's use encryption methods to save cloud logs from different attackers. Authentication access methods are also security strategies used to restrict unauthorized users from illegal access to cloud log files stored in the cloud log storage. However, on access to cloud log storage, an attacker might delete, cloud log files while compromising availability. Confidentiality is not an issue due to encrypted cloud log files neither is integrity due to difficulty in viewing cloud log file data.

Table 11: Cloud log forensics: Vulnerability points

Possible vulnerability points for cloud log attacks	Description	Confidentiality	Integrity	Availability
Log generation	The attack on cloud logs where the logs are generated. It includes virtual machine, application, host, server, and others.	No	No	Yes
Log collection	The attack on the system and resources where logs are collected from various locations in the cloud.	Yes	Yes	Yes
Network	The attack on the network channel between log generation host and log collector system/agent or between log collector agents and log storage resource.	Yes	Yes	Yes
Log storage	The attack on storage resources where logs are stored by the log collector agents and other cloud storage resources.	No	No	Yes
Log analysis	The attacker exploits resources on which log analysis is performed to investigate various vulnerabilities found in logs.	No	N/A	Yes

Log Analysis. Log analysis is the process to perform analysis on cloud log files collected from cloud log storage. Cloud log analysis identifies attackers through analyzing the cloud log files. The attackers want to keep themselves hidden from being investigated, which forces them to attack the log analysis resource/application to remove evidence of their attack. However, in large cloud computing infrastructure, finding exact location where cloud log analysis is performed is a difficult task, which forces attackers to put more efforts in finding an exact location to attack. Decentralized CLF helps investigator to perform analysis in multiple locations and prevents attackers from exploiting cloud log files at the time of analysis. Confidentiality and integrity are not exploited by attackers during their attacks while the availability of the cloud log files is affected based upon their deletion.

6.3 Cloud Log Forensics: Challenges

To analyze different cloud logs collected from various sources in cloud computing is not an easy task [Damshenas *et al.* 2012]. The distributed infrastructure, virtualized environment, multi-tenant resources, huge running applications, millions of cloud users, real-time response (on demand), and a lot of other factors makes CLF very challenging. The state-of-the-art challenges are introduced and explained in subsequent sections with the aim of providing new research areas for researchers and investigating agencies to develop new models, standards, and frameworks for the CLF process. The CLF challenges are accompanied by proposing solutions to help

researchers in resolving the problems. Table 12 highlights state-of-the-art CLF challenges with proposed solutions.

Cloud Log Data as Big Data. As mentioned earlier, generating massive amounts of cloud log data at various sources causes a problem for CLF investigators in analyzing the cloud log data. The problem relates to the concept called big data, i.e. cloud log data volume, variety, and value. The volume indicates the huge amount of cloud log data generated at multiple locations in cloud computing, which causes difficulties for investigators in real-time environments [Zibin *et al.* 2013]. The analysis of huge amounts of cloud logs data to investigate malicious activities performed by an attacker, which are more complex in cloud computing than in traditional log data computing, requires time [Wesley *et al.* 2014]. Cloud computing has to ensure on-demand services in real-time for users, including cloud log analytics. Moreover, security is an issue for huge cloud log data storage at multiple locations in cloud computing [Popa *et al.* 2011]. However, if any parts of the cloud log storage have been exploited by the attacker, then it will affect the entire investigation process resulting in the reduced integrity of the cloud log data. Similarly, a variety of cloud log data from various sources with different log formats makes CLF more difficult in terms of using a single cloud log analytics approach [Oliner *et al.* 2012]. Each cloud logs created at different locations of cloud computing has its own objective for which it has been generated. For instance, cloud network logs are generated to record various patterns of the packet [Spring 2011] while cloud system logs are used to record different state changes. Each cloud log is captured with different types of information, which complicates CLF by treating each cloud log according to different approaches and tactics. The value of cloud log files produces a significant impact on CLF in terms of providing useful information regarding events. For instance, if cloud logs do not provide sufficient value/information regarding an event occurring previously to help investigators in understanding the situation they are useless. The value provided by cloud log files is that they have to ensure the amount of information captured during the logging process is sufficient to investigate or analyze the situation easily.

As the number of cloud users grows rapidly, user interaction with cloud computing increases that creates more cloud log data [Rong *et al.* 2013]. To handle such a massive amount of cloud log data requires a filtering mechanism to record only the data which is crucial for users, including cloud user, CSP, investigators, and others. The system demands an intelligent mechanism to make decisions about recording and analyzing cloud log data in real-time. For instance, data that contains evidence regarding a malicious event should be recorded and analyzed whereas data which does not contain any sort of malicious event should not be recorded and analyzed. However, making a decision about data in real-time is a very difficult and challenging task for the CSP in order to record and generate cloud logs at the various locations of cloud computing. Some of intelligent mechanisms with useful decisions are discussed. For example, patent-pending LogReduce reduces thousands of log events into group of patterns by removing noise data from it. The transaction analytics provide intelligence across a distributed system to collect and analyze the transactional context of log data to decrease compiling time. The outlier detection analyze thousands of log files with a single query to identify outliers in real-time. The predictive analytics predict future violations and malicious behaviors in log files using linear projection models to prevent it before its appearance. Moreover, a standard cloud logs format must be proposed to fulfill all users' requirements and

minimize the complexity for investigators while analyzing cloud log data. Therefore, reduce analysis time will be reduced for investigators as they will only investigate single cloud log format files.

Accessibility of Cloud Logs. The generation of cloud log files in cloud computing environments is not so difficult, rather having access to them with the proper requirement is [Shams *et al.* 2013]. Each cloud log has to be accessed by authorized individuals having a clear objective. For instance, an application developer will require cloud logs of an application to fix bugs in the application code. Similarly, a network administrator requires network logs to determine the flow of packets. Each cloud log has to be accessed by the group of responsible individuals, according to their requirements [Trenwith and Venter 2014]. No other group can access another cloud log without a valid reason and approval from the legal authorities. Each forensic investigator needs to have full access to the required cloud logs for investigating malicious attacks inside the log data. Appropriate access to cloud logs will result in proper CLF. Moreover, in many cases, the CSP does not allow any third party agency or forensic investigator to have access to the cloud logs for security and privacy reasons [Ruan *et al.* 2012]. For example, Amazon does not share load balancing server logs with anyone, which make difficulties for investigators to perform a proper investigation due to the inaccessibility of different cloud log files. The access to the load balancing server logs by the investigators may disclose the working steps of the load balancing algorithm which may be confidential for the Amazon due to its security and other competitive advantages.

The best option for investigators to access cloud logs is to have a well-established relationship with the CSP. The CSP can help investigators in getting access to cloud logs through the legal permission assigned by the court. However, the problem arises when the CSP becomes untrustworthy due to modification of the cloud logs provided to investigators. Data integrity must be ensured by the investigators when they receive cloud logs from the CSP to identify the (original) malicious activities of the attacker that were recorded at the time of cloud log generation. To monitor any biases of the CSP, human intervention must be minimized by developing an automatic mechanism that sends cloud logs to various authorized investigators by verifying them through different hashing mechanisms. Once investigators confirm that the cloud logs received from the CSP are unmodified, they can start their investigations.

Cloud Log Security. Cloud log file security is significant for CLF due to data confidentiality, integrity, and availability (CIA) [Ryan *et al.* 2011b]. The forensic investigator should ensure the data investigated in the cloud log has not being altered by anyone after the generation. The attacker can exploit cloud log files at the cloud log storage where logs are stored and at the cloud network where data is passed from one place to another and similarly at the cloud log analysis server where log data is investigated for malicious actions. Any violation of cloud log management in terms of CIA will affect all of the CLF by producing biased results. Mostly, log-as-a-service providers perform encryption on cloud log files and store them on cloud storage resources [Sundareswaran *et al.* 2012]. However, once an attacker has found the private key to decrypt the cloud log files, they further perform malicious activities such as deleting attack traces, modification to the cloud log data, and others.

To provide the CIA of cloud log files, the CSP must ensure proper access by enforcing individuals to provide passwords at various levels of their access. Similarly,

encrypting cloud log files as well as a cryptographic key will force attackers to put more effort into accessing and modifying the content of cloud logs. The availability of cloud log files can be ensured by keeping replicate copies of different cloud storage resources. However, one has to further guarantee that all replicas of a cloud log file have been synchronized with each other while accessing any of the replicas during the investigation of the cloud log files. To secure various cloud log files from attackers in the cloud is one of the great challenges for investigators while performing CLF.

Decentralized Cloud Logs. In cloud computing, various cloud logs are generated in different layers while being stored on dispersed log analysis servers. Cloud layers such as operating system, application, network, and database have their own log files with different log formats [Shams *et al.* 2013]. Accessing different cloud logs on each layer of a cloud computing environment is a challenging task for cloud forensic investigators in terms of collecting, preserving, analyzing, and recording log data [Shams *et al.* 2013]. Each log on the different layers of cloud computing could provide vital information for the forensic process and must be accessed for significant evidence. However, a single application running on a virtual machine could have multiple logs stored on multiple log analysis servers placed at different clouds slowing down the CLF process due to accessibility, network delays, servers accessed, availability, and others. The investigation of decentralized cloud logs for malicious activities in a real-time situation is challenging.

The central cloud log analysis mechanism requires the managing of all decentralized log analyses by providing complete and accurate results. The analyses performed at distributed cloud log analysis servers in a cloud computing environment must be synchronized with each other in order to investigate the malicious activities of the attacker in the cloud logs by providing on time investigation results. However, the centralized cloud log analysis mechanism would be easy if all distributed cloud log analysis servers running to analyze cloud logs are controlled by a single CSP. The situation becomes more challenging when cloud logs are analyzed for cloud log analysis servers placed at different data centers of different clouds controlled by different CSPs. To synchronize all distributed cloud log analysis servers requires the willingness of all CSPs to make cloud log analysis more manageable and transparent.

Standardized Cloud Log Format. Due to various cloud log files being generated in a cloud computing environment there are many cloud log formats depending on requirements. For instance, cloud application logs have their own log format to record information while cloud network logs have their own format to record packet information. No single standard cloud logs format has been presented yet to represent various cloud logs within a single format [Marty 2011]. The single cloud log format can help investigators easily investigate cloud logs while having full concentration on their main objectives such as cloud log analysis. On the other hand, it is possible to miss some kinds of information in recording cloud logs which might be essential for the identification of malicious activities by an attacker. Therefore, the entire investigation process will become useless due to the incomplete information presented in the cloud logs. Moreover, it is possible that the cloud application log in cloud-1 has one log format while the same cloud application running in another cloud such as cloud-2 uses a different cloud log format. The multiple cloud logs formats for the same cloud application makes the investigation process more ambiguous and complex for investigators analyze cloud log data in a real-time situation. As a result, a standardized cloud log format is essential for conducting accurate and reliable CLF.

An automated single cloud log format approach is required for converting different types of cloud log format to a single format. The single cloud logs format will assist investigators in understanding cloud log data easily and provide accurate results regarding the malicious activities presented in the cloud log data. The aforementioned proposed solution can be implemented more easily when an organization logs only what they believe is important for them. Therefore, log information entries will be reduced and makes easy to automate cloud logging by producing a single log format.

Fairness of Cloud Log Analysis. The main challenge for cloud investigators carrying out CLF is verifying the fairness of the cloud log analysis process. In most cases, cloud log analysis is performed by junior administrative staff as less priority is given to analyzing cloud logs. CSPs place less focus on cloud log analysis due to the belief that it provides few benefits, given the small output while analyzing large amounts of data and taking up a huge amount of time. However, this is not the case. The time spent on investigating cloud logs helps CSPs understand the work flow of the recorded information as well as identify the vulnerabilities recorded inside the cloud logs so as to assist them in detecting and preventing the vulnerabilities in the future. However, how can the cloud user know that the log analysis performed by the CSP is valid, meaning that the analysis is performed without any alteration or modification of the cloud log data? Similarly, how can one verify the analysis performed on the cloud logs is the original one or analysis contains all of the recorded information that was supposed to be present? CLF has to answer the aforementioned questions to ensure the investigation process is fair and clear in front of the cloud user and the court.

Automatic cloud log analysis tools should be developed to analyze cloud log files generated at various sources in cloud computing. If only one individual is involved in conducting cloud log analysis, there are more chances to miss useful information during analysis intentionally or unintentionally making entire investigation process biased? The probability of unfairness in performing cloud log analysis using automatic CLF tools could be minimized by reducing human interference. Similarly, automatic CLF should collect cloud log files from cloud log storage resources while ensuring data integrity through the use of various data security methods.

Table 12: Cloud log forensics: Challenges

Cloud log forensics challenges	Proposed solution	Description
Cloud log data as a big data	- Data filtering mechanism	To record only significant data in the cloud log data file.
Accessibility of cloud logs	- Dependence on cloud service providers	The CSP has to provide cloud logs to different investigators due to their control on various cloud logs. However, data integrity must be ensured by investigators.
Cloud log security	- Proper access methods - Encryption of cloud log files and cryptographic key - Replication of cloud log files	Cloud logs must only be accessed by authorized individuals through different access methods. Both the cloud logs data and encryption key is encrypted due to better and reliable cloud log security. The cloud logs data file is replicated on multiple cloud storage resources
Decentralized cloud logs	- Centralized log analysis	To control and manage entire distributed

Cloud log forensics challenges	Proposed solution	Description
		cloud log analysis servers
Standardized cloud log format	- Single cloud log format	Every cloud log generated at multiple locations in the cloud computing must have a single cloud log format with filled entries according to the requirement.
Fairness of cloud log analysis	- Automatic cloud log analysis tool	A tool used to analyze cloud logs automatically with minimum human interventions.

7. CONCLUSIONS AND FUTURE DIRECTIONS

First, we present the conclusive results derived from the sections of the paper. Then, we present future directions of CLF that guides researchers, CSPs, investigators, legislators, and cloud vendors to work out on these open issues to make CLF more realistic and implementable.

7.1 Conclusions

The integration of cloud logs with digital forensics emerge a new research field, i.e. CLF in cloud computing security. Recently, different research works have been conducted on CLF that come up with its proposed solutions. For instance, a work in [Shams *et al.* 2013] proposed a secure cloud logging architecture that collect information from distributed logs to generate a single image of the operation by providing in-depth investigation. In [Marty 2011], a single log collector and processor are introduced to provide reliable and secure data for investigators in a standardized way. The centralize log management decreases the time overhead for users and organizations. In [Thorpe *et al.* 2011b], synchronize cloud log forensic framework is proposed to reconstruct events in cloud computing based on VM and physical disk log files. The reconstruction of events through logs assists investigators to track malicious behavior of the cloud log attacks. In [Thorpe *et al.* 2013b], hypervisor event logs are used as a source of VM evidence for cloud computing forensics. The temporary inconsistency in VM logs is detected while using activity timelines. Recently, in [Patrascu and Patriciu 2015] a modular layer-based logging framework for cloud computing forensics is proposed to monitor malicious users activities.

Besides all the research efforts done in CLF, still there are various issues which have to be addressed to make a real CLF implementation. A suitable option is to generate logs for each and every event occurring in cloud computing in order to record all malicious behavior. However, cloud logs are generated at different locations resulting in a large number of cloud log files which require proper cloud log management. Cloud log management is essential to ensure that cloud logs are stored on secure resources with adequate information for specific periods of time. Cloud logs benefit forensic investigators in the identification of fraudulent events, security incidents, policy violations, and operational problems. Cloud logs also assist in establishing baselines, performing audit analysis, carrying out internal investigations, identifying long-term problems, and others.

However, the lack of CLF standards makes investigation difficult for investigators. For instance, there is no accessibility policy related to the accessing of cloud log files from cloud computing resources, there is no data integrity mechanism for cloud log files, and an absence of user data privacy in cloud log files, and much more. To overcome the aforementioned CLF problems, cloud log-as-a-service providers have to work on a set of recommendations that include: (a) establishing a

standardized policy and standardized set of procedures, (b) creating and maintaining a separate cloud log management infrastructure, (c) developing secure cloud logging storage, (d) assigning expert manpower to cloud log management, (e) giving priorities to operational cloud logging, (f) developing a standardized operational process for cloud logging, and (g) correlating the distributed cloud logs with a central cloud log management.

7.2 Future Directions

In this section, new future research directions in the context of CLF are presented. However, CLF is still in its early stage of the research to provide ample opportunities for both technical and economic future work to mitigate the challenges related to its paramount log management. Each future direction as shown in Figure 5 will bring the focus of academicians, industrialists, vendors, and CSPs to research out profound solutions for CLF in making them applicable within cloud computing.

Cloud Log Forensics APIs. Nowadays, cloud computing provides different APIs to help clients to interact with cloud resources for different services including storage and computation. However, CLF lacks standardized APIs to assist investigators in accessing cloud log data for analyzing malicious events occurred at the time of the attack. In [Patrascu and Patriciu 2014], cloud forensics API is proposed which used to collect log data from VM placed inside virtualization layer. The cloud forensics API bridge between the investigator and the monitor VM for a specific amount of time to collect different logs. The proposed cloud forensics API lacks to provide log data between different VM which may be vital for VM side channel attacks. Therefore, a novel research work requires in developing unique and secure APIs for CLF to provide easy and secure interfaces for investigators to analyze cloud log data within and outside the cloud. Conversely, if APIs are not properly developed, causing vulnerabilities, will affect entire CLF by harming cloud log data while having spurious access to it. Due to multiple architecture layer of cloud computing, various cloud log storages, numerous accesses to cloud logs, migration of cloud log data, and correlation of cloud log files creates complication in standardizing CLF APIs in cloud computing. To overcome the aforementioned complications, the large players of cloud computing have to take necessitate and immediate actions in developing standardized CLF APIs. However, cloud log data will be in continuous risk due to numerous attacks resulting inadequate investigation output for forensics queries. At this early stage of CLF, it may be difficult for the individual cloud vendor to put more effort for more output. The best option would be to syndicate specific expertise of each cloud vendors by putting fewer efforts for more output such as producing standards CLF APIs. Such effort will reduce time in proposing and developing globally accepted standardized CLF APIs. As a result, cloud users will feel more comfort in accessing their cloud log data, whereas accessing through secure CLF APIs in the cloud. Therefore, new cloud logs forensics APIs are demanded for comprehensive and accurate investigation of cloud log data.

Chain of Custody. Chain of Custody (CoC) refers to record sequential states during an event without losing any information due to modification, deletion, and insertion. The CoC is important to understand the entire process by connecting each event to another for extracting useful information. In CLF, CoC is defined as different attributes including verifiable evidence, log locations, log storage positions, log access methods, and collecting process of logs that explain and verify each step i.e. from collecting of log files to presenting log evidence in the court. In general, CoC

should ensure in cloud computing that how was log files created, stored, analyzed, and presented in the court. In cloud computing, it is very difficult to perform each step of CLF due to resource in-accessibilities, geographical diversification, virtualization, multiple layer architecture, and millions of users. Most of the time, cloud logs are generated by the CSP are restricted from third party investigators to have access because of their own corporative security laws and procedures. The challenge arises when an investigator what to verify its CLF steps against the culprit in the court. A question should be raised against CLF CoC that how much CSP was trustworthy to provide cloud log evidences to the investigator. A comprehensive law, procedures, and standards should be created with consultation of CSPs and investigators to have clear and true CoC procedure for each step of CLF. The CoC is considered one of the important future directions of CLF due to its significance in terms of verifiability, understandability, and dependability of the whole process.

Metadata Support Forensics. Metadata of cloud logs plays a vital role in providing supportive evidence of the crime scene in cloud computing. The metadata of cloud logs may include log file creation, access, modification, resource shift, and its size. Metadata information provides useful insight to investigator in analyzing cloud logs easily. But time to time, these metadata information changes due to migration of cloud log files and multiple accesses to the cloud logs. For instance, a cloud log file was created at a specific time and its metadata was stored with its current information status. Later, if cloud log file is migrated to another resource within same cloud or another cloud, it will change the metadata information due to its afterward access, migration, and log formation. Similarly, in case of multiple accesses to the cloud log file by multiple users, metadata information about the cloud log is changed which creates biasness in the CLF in terms of tracing exact individual/user responsible for accessing cloud logs. The metadata of cloud log file may retrieve by the investigator for analyzing data while it has been accessed or modified later by another individual to create inaccurate evidence regarding investigating a crime scene in cloud computing. However, most of the time, metadata is not a concern of an attack by the attacker, but it can be altered by him for the sake of hiding himself. Therefore, as a future direction, metadata has to analyze in-depth by generating standard policies, procedures, and law, especially in terms of cloud logs migration to multiple cloud resources in another cloud and keeping track of its values every time. The metadata has to be kept secure enough so that it cannot be altered by any unauthorized individual. In [Thorpe *et al.* 2012a], kernel hypervisor logs of the VM operating system are overviewed that provide metadata information for cloud log forensics. However, no indication is provided in the paper to know how metadata should be obtained from kernel hypervisor logs while not having access to the complete cloud system. Therefore, a novel research is required to extract useful information from disperse metadata present in distributed cloud environment. An efficient data mining techniques require to efficiently retrieve useful information from huge metadata set of cloud logs with a real-time response.

Selection of an Appropriate Entity (CSP Vs Investigator). Most of cloud resources within the territory of cloud computing are in control of CSP i.e. due to its ownership. In the process of investigating cloud logs, an investigator requires to have an access to cloud log data for analyzing malicious events. The requirement becomes necessary when threats have to be investigated in a real-time due to severe risk of the attack. The challenge arises when an attack has to be investigated by the investigator in cloud log data in real-time, whereas the only access to cloud logs are

with CSP. A proper forensics response management requires identifying intelligently the scope of investigation and performing an immediate action to contact CSP or cloud log investigator. For investigation queries, CSP can get cloud log data to initially analyze the situation immediately based on its investigation capabilities rather than sending data to the investigator which will consume a lot of time. However, in most of the cases, an expert investigator requires to investigate cloud logs for malicious events which cannot be analyzed by the CSP. The decision to identify a responsible entity (CSP or investigator) to have an access and investigate cloud log data requires an understandability of attack behavior and situation. Currently, research work lacks to address the aforementioned issue, which has to be utterly addressed to make CLF works efficient. As a result, a trust level has to be created between CSP and investigator to overwhelm the challenge of identifying and selecting appropriate player for investigating cloud logs at immediate response. For instance, CSP has to ensure the expertise of an investigator that can investigate cloud log data easily and accurately, whereas, investigator should have clear understandability and knowledge about the infrastructure of cloud computing and cloud log management for performing proper investigation. In the literature, trust models and platforms have been proposed for cloud computing with different objectives rather than focusing a level of trust between investigators and CSPs. For instance, in [Ahmad *et al.* 2012], three level trust model has been proposed between users and CSP. In the first two levels, the user has to fulfill the satisfaction constraints, so that it can trust CSP in the third level of trust models. At first level, user should be satisfied from previous experience of the CSP. At second level, the user must be aware completely of the SLA. When these two levels of trust are satisfied, the user can trust the CSP. In [Shen *et al.* 2010], trusted computing platform is integrated with cloud computing architecture to provide confidentiality, integrity, and authenticity. The proposed platform provides benefit for rule-base access and data protection schemes in cloud computing. In [Santos *et al.* 2009], trusted cloud computing platform is proposed for IaaS to provide a closed box execution environment to execute guest VM before it is formally requested by the user. It assists users to verify a secure execution environment provided by the CSP.

Correlation of Cloud Logs. Transparent management of cloud computing conceal execution of an application from the user with an aim to provide a simple interface for usage. Similarly, cloud logs of user applications running on the cloud resources are concealed from cloud users, information includes what, when, where, and how is logging performed. In cloud computing, a log can be created in one cloud whereas, it is stored in another cloud. Likewise, one application has more than one log file, store on more than one cloud resource, disperse within cloud computing. It has more probability that each cloud log file may have different log format and time record. The different log format and time record creates a challenge to correlate different cloud log files of a same application store at different cloud resources. Time synchronization within a cloud logs is a great challenge for forensics, especially in cloud computing. In [Lemoudden *et al.* 2014], a vertical layer “audit & monitor center” is proposed to monitor horizontal layers of the cloud computing in providing a correlation between cloud logs. The audit & monitor center provides a unique identifier to different components in cloud computing infrastructure in a logical and standardized way to keep real-time identifier updates for correlation purposes including correlation of cloud logs as well. The assigned identifiers and centralized log management, consolidate cloud logs from different parts of the cloud infrastructure in a real-time. However, a novel research work requires overcoming

the correlation of cloud logs problem by developing globally accepted standard laws, policies, and procedures. Trustful interfaces require to be created among CSP to exchange cloud log updates seamlessly through secure communication channels. However, there is no legal standard available till now that has to be followed by CSPs for log information exchange. The problem of correlating cloud logs has to be addressed to conduct fair and sound CLF for investigating malicious events and produce accurate results to cloud users.

Real-time Cloud Log Forensics Visualization. The in-depth execution detail of an application execution is hidden from cloud users due to its complexity on cloud resource. Each cloud user view its application process simply as an interface interaction with cloud whereas, actual execution steps are performed seamlessly. Similarly, CLF is performed on cloud logs generated from various locations such as user applications, networks, systems, resources, and security devices without providing detail execution information on its investigation steps to cloud users. Now a day, cloud user is more intent to know each and every event related to its data inside cloud computing. As a result, CLF should ensure the legitimate user data is not being accessed or modified during any investigation steps for analyzing cloud log data. The best option is to record each investigation steps and present it in a visualize form. The Logentries cloud log service provider offers visualization for log analysis management. The visualization provides instant visibility to users by providing in-depth information regarding log files store in the data centers of a cloud. The easy to use dashboard provides user to interact various cloud log related data in getting detailed understandability and information related to cloud log analysis. However, visualization completely CLF steps in a real-time is a great challenge due to distributed cloud infrastructure, multiple cloud log storages, lack of cloud log correlation, and undeveloped CLF tools. The visualization of CLF will make the investigation process simple and understandable to cloud user and will drive decision for future actions. Therefore, increase amount of cloud log data generated at cloud computing requires a visualization tool to provide predictive, description, and prescription analytics for cloud log data, helping investigators in a real-time investigation.

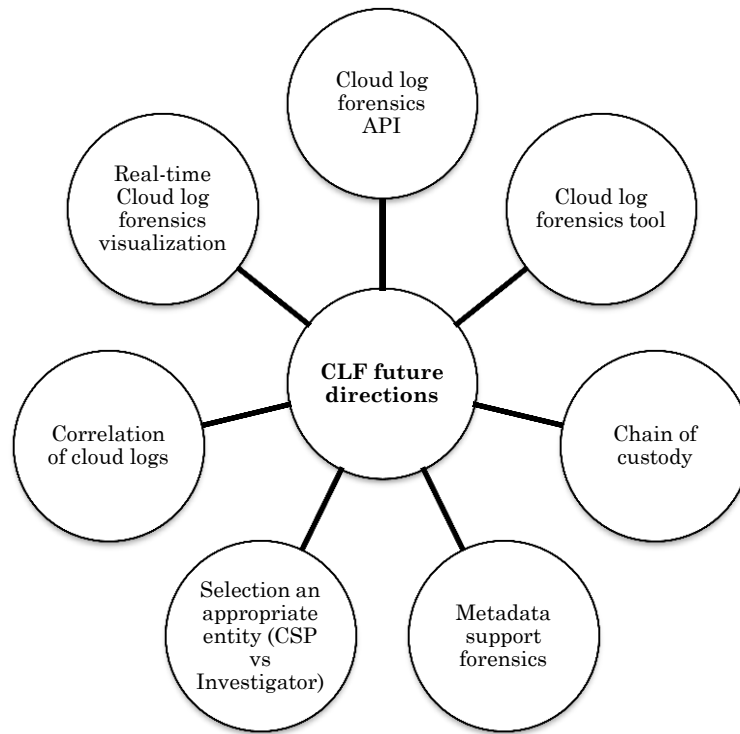


Fig. 5. Future directions for cloud log forensics

Cloud Log Forensics Tools. Log data are considered one of important evidence against malicious attacks during attack investigation in cloud computing. The log data inside cloud log files placed on distributed cloud resources has to be analyzed in a real-time, which is of a great challenge. To perform analytics on cloud log data, an automatic CLF tool is required to collect cloud log files from distributed locations and investigate them for extracting valuable evidence. In [Thorpe *et al.* 2011a], Virtual Machine Log Auditor (VMLA) is proposed as a cloud log forensics tool to provide a graphical interface for timelines of virtual machine (VM) hypervisor log events gathered from different physical operating systems. The VMLA primary objective is to assist the investigator to know what VM events including modification, access, and creation happened in the physical operating system. However, till date, no standardized CLF tool is developed to collect and analyze cloud log files placed on different cloud resources within and another cloud. The hurdles to develop CLF tools ascends due to layer infrastructure, distribution and virtualize environment, numerous resources, share networks and resources, millions of users, and centralized control of cloud computing. To overcome the aforementioned hurdles, industry professionals have to make coordination with CSP and legal personnel to develop new CLF tools without violating service level agreements between the cloud user and CSP as well as jurisdiction laws. One option to develop CLF quickly is to propose an open source CLF tool, where professionals from worldwide will contribute different modules and functionality to it. Similarly, cloud log investigators have to suggest their opinions to cloud professional regarding their perception related to the ideal CLF tool. At the end, cloud log investigators would be the one using tool for analyzing different cloud logs in cloud computing. Therefore, the need for highly standardized

CLF tools is of high importance to investigate different cloud logs in cloud computing in a real-time.

ACKNOWLEDGMENTS

This work is funded by Bright Spark Unit, University of Malaya, Malaysia and High Impact Research Grant UM.C/625/1/HIR/MOE/FCSIT/17, Malaysian Ministry of Higher Education under the University of Malaya. Co-author Buyya's work is supported by a Future Fellowship by the Australian Research Council.

REFERENCES

- A. Burton. 2014. Real-time log management and analytics at any scale. (2014). Retrieved November 16th, 2015 from <https://logentries.com/>
- A. Chuvakin, K. Schmidt, and Chris Phillips. 2013. Chapter 2 - what is a log?. Syngress. (2013).
- A. Gani, G. M. Nayeem, M. Shiraz, M. Sookhak, M. Whaiduzzaman, and S. Khan. 2014. A review on interworking and mobility techniques for seamless connectivity in mobile cloud computing. *Journal of Network and Computer Applications* 43 (2014). 84-102.
- A. Holovaty. 2014. Django Makes It Easier to Build Better Web Apps More Quickly and with Less Code. (2014). Retrieved November 16th, 2015 from <https://www.djangoproject.com/>
- A. Oliner, A. Ganapathi, and W. Xu. 2012. Advances and challenges in log analysis. *Communications of the ACM* 55, 2 (2012), 55-61.
- A. Patrascu, and V. V. Patriciu. 2014. Logging framework for cloud computing forensic environments. In *Proceeding of the IEEE 10th International Conference on Communications (COMM)*. 1-4.
- A. Patrascu and V. V. Patriciu. 2015. Logging for cloud computing forensic systems. *International Journal of Computers Communications & Control* 10, 2 (2015). 222-229.
- A. Prasad, and P. Chakrabarti. 2014. Extending access management to maintain audit logs in cloud computing. *International Journal of Advanced Computer Science and Applications* 5, 3 (2014). 144-147.
- A. Rafael. 2013. Secure log architecture to support remote auditing. *Mathematical and Computer Modelling* 57, 7 (2013), 1578-1591.
- A. Stanojevic. 2013. Banca Intesa counters threats with HP ArcSight. Case Study. Hewlett-Packard. 4 pages. Retrieved November 16th, 2015 from <http://www8.hp.com/h20195/V2/GetPDF.aspx/4AA4-6020ENUS.pdf>
- A. Williams. 2013. Loggly, a Splunk Competitor, Raises \$10.5m for Cloud-Centric Approach to Log Management. (2013). Retrieved November 16th, 2015 from <http://techcrunch.com/2013/09/03/loggly-a-splunk-competitor-raises-10-5m-for-cloud-centric-approach-to-log-management/>
- Amazon. 2015. Amazon Simple Notification Service. (2015). Retrieved November 16th, 2015 from <http://aws.amazon.com/sns/>.
- B. Mizerany. 2014. Put this in your pipe and smoke it. (2014). Retrieved November 16th, 2015 from <http://www.sinatrarb.com/>
- B. Mollamustafaoglu. 2014. We make alerts work for you. (2014). Retrieved November 16th, 2015 from <https://www.opsgenie.com/>
- B. R. Carrier. 2006. Risks of Live Digital Forensic Analysis. *Communications of the ACM* 49, 2 (2006), 56-61.
- C. C. Yun, J. Y. C. Chang, B. B. C. Chiu, D. Y. Shue, Y. Kaneyasu, and J. W. Warfield. 2014. Ensuring integrity of security event log upon download and delete. (2014), *U.S. Patent* No. 8,856,086.
- C. Oppenheimer. 2009. Loggly reveals what matters. (2009). Retrieved November 16th, 2015 from <https://www.loggly.com/>
- C. Rong, S. T. Nguyen, and M. G. Jaatun. 2013. Beyond lightning: a survey on security challenges in cloud computing. *Computers & Electrical Engineering* 39, 1 (2013), 47-54.
- D. J. Scales, M. Xu, and M. D. Ginzton. 2013. *Low overhead fault tolerance through hybrid checkpointing and replay*. *U.S. Patent* No. 8,499,297 (2013).
- D. Birk. 2011. Technical challenges of forensic investigations in cloud computing environments. In *Workshop on Cryptography and Security in Clouds*. 1-6.
- D. Birk, and C. Wegener. 2011. Technical issues of forensic investigations in cloud computing environments. In *Proceeding of the IEEE 6th International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE)*. 1-10.
- E. Casey. 2009. Handbook of Digital Forensics and Investigation. Academic press (2009).
- E. J. Janger and P. M. Schwartz. 2001. Gramm-Leach-Bliley Act, information privacy, and the limits of default rules. *The Minn. L. Rev.* 86 (2001), 1219.
- E. Lindvall. 2014. How Papertrail makes life easier. (2014). Retrieved November 16th, 2015 from <https://papertrailapp.com/>

- G. Rocher. 2005. A powerful Groovy-based Web application framework for the JVM. (2005). Retrieved November 16th, 2015 from <https://grails.org/>
- G. Samudra. 2005. Extending Log4j to create custom logging components. In *Logging in Java with the JDK 1.4 logging API and Apache log4j*. Apress. 235-284.
- H. A. Jahdali, A. Albatli, P. Garraghan, P. Townend, L. Lau, and Jie Xu. 2014. Multi-tenancy in cloud computing. In *Proceeding of the IEEE 8th International Symposium on Service Oriented System Engineering*. 344-351.
- H. Chung, J. Park, S. Lee, and C. Kang. 2012. Digital Forensic Investigation of Cloud Storage Services. *Digital Investigation* 9, 2 (2012), 81-95.
- H. H. Mao, C. J. Wu, E. E. Papalexakis, C. Faloutsos, K. C. Lee, and T. C. Kao. 2014. MalSpot: multi2 malicious network behavior patterns analysis. In *Advances in Knowledge Discovery and Data Mining*. Springer International Publishing (2014). 1-14.
- I. M. Abbadi. 2014. *Cloud Management and Security*. John Wiley & Sons (2014).
- I. Ray, K. Belyaev, M. Strizhov, D. Mulamba, and M. Rajaram. 2013. Secure logging as a service---delegating log management to the cloud. *IEEE Systems Journal* 7 (2013), 323-334.
- J. Dykstra and A. T. Sherman. 2011. Understanding issues in cloud forensics: two hypothetical case studies. *Journal of Network Forensics* 3, 1 (2011). 19-31.
- J. Gerring. 2007. *Case study research. Principles and Practices*. Cambridge University Press (2007).
- J. Hash, P. Bowen, A. Johnson, C. D. Smith, and D. I. Steinberg. 2008. An introductory resource guide for implementing the health insurance portability and accountability act (HIPAA) security rule. Doctoral dissertation, *National Institute of Standards and Technology* (2008).
- J. H. Beaver. 2015. Lessons on Efficient Log Analysis from Monex Insight. Case Study Report. Loggly Research. 3 pages. <https://www.loggly.com/blog/lessons-efficient-log-analysis-monex-insight/>
- J. Sissel. 2014. Process any data, from any source. (2014). Retrieved November 16th, 2015 from <https://www.elastic.co/products/logstash>
- J. South. 2013. Heartland payment systems hardens applications and blocks attacks with the aid of HP security software. Technical Report. IDC Go-To-Market Services. 4 pages. <http://www8.hp.com/h20195/V2/GetPDF.aspx/4AA5-1356ENW.pdf>
- J. Spring. 2011. Monitoring cloud computing by layer, part 1. *IEEE Security & Privacy* 9, 2 (2011), 66-68.
- J. Stoppelman. 2004. AWS Case Study: Yelp. Case Study. Amazon. Retrieved November 16th, 2015 from <https://aws.amazon.com/solutions/case-studies/yelp/>
- J. T. Force and T. Initiative. 2013. Security and privacy controls for federal information systems and organizations. NIST Special Publication 800 (2013). 53.
- J. Turnbull. 2005. Understanding logging and log monitoring. *Hardening Linux. A-press* (2005). 233-280.
- J. W. Joo, J. H. Park, S. K. Suk, and D. G. Lee. 2014. LISS: log data integrity support scheme for reliable log analysis of osp. *Journal of Convergence* 5, 4 (2014).
- J. Wei, Y. Zhao, K. Jiang, R. Xie, and Y. Jin. 2011. Analysis farm: a cloud-based scalable aggregation and query platform for network log analysis. In *Proceedings of the IEEE International Conference on Cloud and Service Computing (CSC)*. 354-359.
- J. Yang, N. Plasson, G. Gillis, N. Talagala, and S. Sundararaman. 2014. Don't stack your log on my log. In *USENIX Workshop on Interactions of NVM/Flash with Operating Systems and Workloads (INFLOW)*. (2014).
- J. Yin. 2014. Cloud based logging service. *US Patent* 20,140,366,118 (2014).
- K. Kent, S. Chevalier, T. Grance, and H. Dang. 2006. Guide to Integrating Forensic Techniques into Incident Response. *NIST Special Publication*. (2006). 800-886.
- K. Kent, and M. Souppaya. 2014. Guide to computer security log management. National Institute Of Standards and Technology (2014).
- K. L. K. Ryan, P. Jagadpramana, and B. S. Lee. 2011a. Flogger: a file-centric logger for monitoring file access and transfers within cloud computing environments. In *Proceeding of the International Joint Conference of IEEE TrustCom-11/IEEE ICSS-11/FCST-11*. 765-771.
- K. L. K. Ryan, M. Kirchberg, and B. S. Lee. 2011b. From system-centric to data-centric logging-accountability, trust & security in cloud computing. In *Proceeding of the IEEE Defense Science Research Conference and Expo (DSR)*. 1-4.
- K. Popovic and Z. Hocenski. 2010. Cloud computing security issues and challenges. In *Proceedings of the IEEE 33rd International Convention (MIPRO)*. 344-349.
- K. Ruan, J. Carthy, T. Kechadi, and M. Crosbie. 2011. Cloud forensics, *Advances in digital forensics VII*. Springer Berlin Heidelberg (2011). 35-46.
- K. Ruan, J. James, J. Carthy, and T. Kechadi. 2012. Key terms for service level agreements to support cloud forensics. *Advances in digital forensics VIII*. Springer Berlin Heidelberg (2012). 201-212.
- K. Saurabh and C. Beedgen. 2014. Master your data continuous intelligence. (2014). Retrieved November 16th, 2015 from <https://www.sumologic.com/>

- M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. 2010. A view of cloud computing. *Communications of the ACM* 53, 4 (2010), 50-58.
- M. Baum. 2014. Analyze & troubleshoot your cloud applications. Technical Report. SplunkStorm. 2 pages. https://www.splunk.com/web_assets/pdfs/secure/Storm_Product_Fact_Sheet.pdf
- M. Bradley and A. Dent. 2010. Payment Card Industry Data Security: What it is and its impact on retail merchants. Technical Report. Royal Holloway Series. 15 pages. http://cdn.ttgtmedia.com/searchsecurityuk/downloads/RHUL_Bradley_2010.pdf
- M. Damshenas, A. Dehghantanha, R. Mahmoud, and S. B. Shamsuddin. 2012. Forensics investigation challenges in cloud computing environments. In *Proceeding of the IEEE International Conference on Cyber Security, Cyber Warfare and Digital Forensics (CyberSec)*. 190-194.
- M. Ellis. 2013. IBM Operations Analytics-Log Analysis. (2013). Retrieved November 16th, 2015 from <http://www-03.ibm.com/software/products/en/ibm-operations-analytics---log-analysis>
- M. Lemoudden, N. Bouazza, and B. E. Ouahidi. 2014. Towards Achieving Discernment and Correlation in Cloud Logging. In *Proceeding of the Applications of Information Systems in Engineering and Bioscience*. 202-207.
- M. Sato, and T. Yamauchi. 2013. Secure log transfer by replacing a library in a virtual machine. In *Advances in Information and Computer Security*. Springer Berlin Heidelberg (2013). 1-18.
- M. Shiraz, A. Gani, A. Shamim, S. Khan, and R. W. Ahmad. 2015. Energy efficient computational offloading framework for mobile cloud computing. *Journal of Grid Computing* 13, 1 (2015), 1-18.
- M. Taylor, J. Haggerty, D. Gresty, and D. Lamb. 2011. Forensic investigation of cloud computing systems. *Network Security* 2011, 3 (2011), 4-10.
- M. Vrable, S. Savage, and G. M. Voelker. 2012. BlueSky: a cloud-backed file system for the enterprise. In *Proceedings of the 10th USENIX conference on File and Storage Technologies*. 19-19 (2012).
- N. Prabha, C. Timotta, T. Rajan, and A. Jaleef PK. 2014. Encrypted query processing based log management in the cloud for improved potential for confidentiality. *International Journal of Computer Applications Technology and Research* 3, 5. (2014). 309-311.
- N. Santos, K. P. Gummadi, and R. Rodrigues. 2009. Towards Trusted Cloud Computing. In *Proceedings of the 2009 conference on Hot topics in cloud computing*. 3-3.
- P. Heath. 2014. Monitor your apps every single second. (2014). Retrieved November 16th, 2015 from <http://www.bmc.com/truesightpulse/customers/>
- P. M. Trenwith and H. S. Venter. 2014. A digital forensic model for providing better data provenance in the cloud. In *Proceedings of the IEEE Information Security for South Africa (ISSA)*. 1-6.
- P. Mell and T. Grace. 2011. The NIST definition of cloud computing. NIST Special Publication 800-145 (2011).
- Q. Han, M. Shiraz, A. Gani, M. Whaiduzzaman, and S. Khan. 2014. Sierpinski triangle based data center architecture in cloud computing. *The Journal of Supercomputing* 69, 2 (2014), 887-907.
- R. A. Popa, J. R. Lorch, D. Molnar, H. J. Wang, and L. Zhuang. 2011. Enabling security in cloud storage SLAs with cloudproof. In *Usenix Annual Technical Conference*. 242 (2011).
- R. Buyya, C. S. Yeo, and S. Venugopalirk. 2008. Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities. In *Proceeding of the IEEE 10th International Conference on High Performance Computing and Communications*. 5-13.
- R. Buyya, C. S. Yeo, S. Venugopalirk, J. Broberg, and I. Brandic. 2009. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems* 25, 6 (2009), 599-616.
- R. Dahl. 2014. Node.js on the Road. (2014). Retrieved November 16th, 2015 from <https://www.joyent.com/noderoad>
- R. Marty. 2011. Cloud application logging for forensics. In *Proceedings of the 2011 ACM Symposium on Applied Computing*. 178-184.
- R. Vaarandi and M. Pihelgas. 2014. Using security logs for collecting and reporting technical security metrics. In *Proceedings of the IEEE Military Communications Conference (MILCOM)*. 294-299.
- S. Ahmad, B. Ahmad, S. M. Saqib, and R. M. Khattak. 2012. Trust Model: Cloud's Provider and Cloud's User. *International Journal of Advanced Science and Technology* 44, (2012), 69-80.
- S. Butterfield, E. Costello, C. Henderson, and S. Mourachov. 2014. Slack so yeah, we tried slack. (2014). Retrieved November 16th, 2015 from <https://slack.com/>
- S. Khan, A. Gani, A.W. A. Wahab, and M. A. Bagiwa. 2015. SIDNFF: Source Identification Network Forensics Framework for Cloud Computing. In *Proceeding of the IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*. 418-419.
- S. Khan, E. Ahmad, M. Shiraz, A. Gani, A. W. A. Wahab, and M. A. Bagiwa. 2014a. Forensic challenges in mobile cloud computing. In *Proceeding of the IEEE International Conference on Computer, Communication, and Control Technology (IACCT 2014)*. 343-347.

- S. Khan, K. Hayat, S. A. Madani, S. U. Khan, and J. Kolodziej. 2012. The median resource failure check pointing. In 26th European Conference on Modelling and Simulation (ECMS). 483-489.
- S. Khan, M. Shiraz, A. W. A. Wahab, A. Gani, Q. Han, and Z. B. A. Rahman. 2014b. A comprehensive review on adaptability of network forensics frameworks for mobile cloud computing. *The Scientific World Journal* 2014, 547062 (2014). 27.
- S. Ramgovind, M. M. Eloff, and E. Smith. 2010. The management of security in cloud computing. In *Proceeding of the IEEE Information Security for South Africa (ISSA)*. 1-7.
- S. Simou, C. Kalloniatis, E. Kavakli, and S. Gritzalis. 2014. Cloud Forensics: Identifying the major issues and challenges. In *Advanced Information Systems Engineering*. Springer International Publishing (2014). 271-284.
- S. Sundareswaran, A. C. Squicciarini, and D. Lin. 2012. Ensuring distributed accountability for data sharing in the cloud. *IEEE Transactions on Dependable And Secure Computing*, 9, 4 (2012). 556-568.
- S. T. On, J. Xu, B. Choi, H. Hu, and B. He. 2012. Flag commit: supporting efficient transaction recovery in flash-based dbms. *IEEE Transactions on Knowledge and Data Engineering*, 24, 9 (2012), 1624-1639.
- S. Thorpe, I. Ray, T. Grandison, and A. Barbir. 2011a. The virtual machine log auditor. In *Proceeding of the IEEE 1st International Workshop on Security and Forensics in Communication Systems*. (2011).
- S. Thorpe, I. Ray, and T. Grandison. 2011b. A synchronized log cloud forensic framework. *The International Conference on Cybercrime, Security & Digital Forensics*. (2011).
- S. Thorpe, I. Ray, and T. Grandison. 2011c. Enforcing data quality rules for a synchronized VM log audit environment using transformation mapping techniques. In *Computational Intelligence in Security for Information Systems*. Springer Berlin Heidelberg. 265-271.
- S. Thorpe, I. Ray, T. Grandison, and A. Barbir. 2012a. Cloud log forensics metadata analysis. In *Proceedings of the IEEE Computer Software and Applications Conference Workshops (COMPSACW)*. 194-199.
- S. Thorpe, I. Ray, T. Grandison, A. Barbir, and R. France. 2013b. Hypervisor event logs as a source of consistent virtual machine evidence for forensic cloud investigations. In *Data and Applications Security and Privacy XXVII*. Springer Berlin Heidelberg. 97-112.
- S. Thorpe, I. Ray, I. Ray, and T. Grandison. 2011d. A formal temporal log data model for the global synchronized virtual machine environment. *International Journal of Information Assurance and Security* 6, 2 (2011). 398-406.
- S. Thorpe, I. Ray, I. Ray, T. Grandison, A. Barbir, and R. France. 2012b. Formal parameterization of log synchronization events within a distributed forensic compute cloud database environment. In *Digital Forensics and Cyber Crime*. Springer Berlin Heidelberg. 156-171.
- S. Thorpe, T. Grandison, A. Campbell, J. Williams, K. Burrell, and I. Ray. 2013a. Towards a forensic-based service oriented architecture framework for auditing of cloud logs. In *Proceeding of the IEEE 9th World Congress on Services*. 75-83.
- T. Nielsen. 2014. Everything you need to build, run, and scale. (2014). Retrieved November 16th, 2015 from <https://www.heroku.com/>
- T. R. Wyatt. 2009. Mission: Messaging: Circular Logs Vs Linear Logs. (2014). Retrieved November 16th, 2015 from http://www.ibm.com/developerworks/websphere/techjournal/0904_mismes.html.
- T. Sang. 2013. A log-based approach to make digital forensics easier on cloud computing. In *Proceeding of the IEEE 3rd International Conference on Intelligent System Design and Engineering Applications (ISDEA)*. 91-94.
- T. Simon. 2014. KPI Dashboards that put your data to work. (2014). Retrieved November 16th, 2015 from <https://www.geckoboard.com/>
- U. Flegel. 2002. Pseudonymizing Unix log files, In *Infrastructure Security*. Springer Berlin Heidelberg. 162-179.
- V. Wesley, T. Harris, L. Long Jr., and R. Green. 2014. Hypervisor security in cloud computing systems. *ACM Computing. Surveys* (2014), 1-22.
- X. Lin, P. Wang, and B. Wu. 2013. Log analysis in cloud computing environment with hadoop and spark. In *Proceeding of the IEEE 5th International Conference on Broadband Network & Multimedia Technology (IC-BNMT2013)*. 273-276.
- Z. Nik. 2011. Detection of network security breaches based on analysis of network record logs. *U.S. Patent No. 7,904,479* (2011).
- Z. Shams, A. K. Dutta, and R. Hasan. 2013. SecLaaS: secure logging-as-a-service for cloud forensics. In *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*. 219-230.
- Z. Shams, M. Mernik, and R. Hasan. 2014. Towards building a forensics aware language for secure logging. *Computer Science and Information Systems*, 11, 4 (2014), 1291-1314.
- Z. Shen, L. Li, F. Yan, and X. Wu. 2010. Cloud Computing System based on Trusted Computing Platform. In *Proceeding of the IEEE Intelligent Computation Technology and Automation (ICICTA)*. 942-945.
- Z. Zibin, J. Zhu, and M. R. Lyu. 2013. Service-generated big data and big data-as-a-service: an overview. In *Proceedings of the IEEE International Congress on Big Data (BigData Congress)*. 403-410.