Managing Cloud-generated Logs Using Big Data Technologies

Mouad Lemoudden, Bouabid El Ouahidi
Mohammed-V University, Faculty of Sciences, L.R.I.
B.O. 1014, Rabat, Morocco
mouad.lemoudden@gmail.com, bouabid.ouahidi@gmail.com

Abstract— **Cloud computing is a fast-growing paradigm that has forcefully emerged and established itself in the next generation of IT industry and business, performing massive-scale and complex computing. As cloud technology adoption continues to progress, massive growth in the scale of data generated through cloud computing has been observed. A large part of that data has been machine-generated log data; it is generated at every layer and component within the cloud information technology ecosystem that spans a range of IT functions from storage and computation to networking and application services. Log files are essential for enterprise-level monitoring, troubleshooting, security, debugging, compliance, etc.**

We argue that cloud computing is in need of a new methodology for recording, managing, storing and analyzing the log data that is generated by the cloud infrastructure, based on big data technologies, allowing for the access to a wealth of knowledge that can contribute to its own advancement. Our paper discusses and expands existing standard log management solutions to design new methods of exploiting the unstructured log data in order to gain unprecedented insights.

Keywords - cloud computing; logs; big data; text analytics; logging; unstructured data; log management

I. MOTIVATION

Cloud computing, evolved from the concept of utility or grid computing [3], offers infinite infrastructure resources, very convenient pay-as-you-go service, and low cost computing. Using virtualization technology, resources are pooled and shared between multiple tenants, allowing applications to run on physical systems that are not specified, data is stored in locations that are unknown, administration of systems is outsourced to others, and access by users is ubiquitous [2]. According to the National Institute of Standards and Technology (NIST) definition of cloud computing, some of the essential characteristics of a cloud data center include on-demand self service, broad network access, resource pooling, rapid elasticity and measured service [1].

In one of their recent reports, Market Research Media stated that the global cloud computing market is expected to grow at an 30% CAGR (compound annual growth rate) to reach $270 billion in 2020 [4].

Cloud computing offers a new horizon for business outlets and IT organizations. However, as cloud technology adoption continues to progress, massive growth in the scale of data generated has been observed. A large part of that data resides in machine-generated log data. Being generated at every layer and component within the cloud information technology ecosystem, log data gives rise to the ability to perform security and auditing tasks through digital forensics. An annual report of the Federal Bureau of Investigation (FBI) claims that the size of the average digital forensic case is growing 35% per year in the United States. From 2003 to 2007, it increased from 83GB to 277 GB [5]. As a consequence, experts have been working on log management tools and techniques to make gain new helpful knowledge. Digital forensic procedures start by collecting evidence. In the cloud, the evidence could very well be the logs generated by the service provider. There are several log management tools available in the market. Unfortunately, they aren't suitable for the dynamic nature of the cloud. Much of the traditional expectation concerning log data (for example, physical access to hardware) are not valid when taking into account the characteristics of cloud computing.

Logging in the Cloud is substantially different than traditional and distributed system logging. The ability to identify sources of irregular results is powered by continuous and frequent logging. Logging is typically done using virtual machine storage that is preserved only for the lifetime of the instance [6]. Cloud resources are shared by definition and it is demanded to ensure that logs are being generated by both VMs and virtualization infrastructure components [7]. Thus, there must be measures to generate and preserve the logs for a postmortem analysis.

The volume and variety of cloud-generated log files is experiencing a massive rise in size that it becomes increasingly difficult for log management solutions to store log files, parse them, and trace potential issues and errors. Log files are essentially a large volume of unstructured data that gets created from various cloud components and resources. Therefore, such growth demands the usage of Big Data techniques for storing, processing and analysis.

By considering log data as an important part of enterprise data supply and exploiting it using Big Data techniques, organizations can now gain access to a wealth of insight residing in unstructured log files, which is presently being collected but underutilized because of its vast variety and volume. Harnessing the power of big data, and taking into account its rise in cloud computing [23], can propel enterprise decision making to new heights, no longer limited to reactive IT operations.

978-1-4673-8224-3/15/$31.00 ©2015 IEEE

This paper will discuss traditional logging and log management, and the challenges that arise in cloud computing territory in section 2. Section 3 motivates the use of Big Data techniques to manage Cloud log data using cloud resources. In section 4, we provide a proposed methodology in order to allow treatment of diverse log format in massive sizes, undocumented proprietary log files that resist traditional analysis techniques, presence of false log records. The last section contains conclusion and perspectives.

II. Log Definition and Cloud Logs

As defined by NIST, "a log is a record of the events occurring within an organization's systems and networks. [9]". To the general public, the log file wouldn't be considered a stimulating area in technology today. However, the growing log data generated for a range of sources can drive large business transformations when combined with state of the art analytics.

A. Log Data and Analysis

Log data resides in log messages that are generated by an IT component in response to stimuli, depending on the source of the log file (login, logout, failures, etc). Log messages can be classified as informational, Debug, Warning, Error, and Alert [8]. Traditionally, a loghost is centralized computer system where log messages are collected from multiple locations. A loghost is also used to store backup copies of logs and to perform analysis on log data. Some of the more common protocols used to transfer logs are:

- Syslog: UDP based client/server protocol. It is the most used and prevalent logging method.
- SNMP: it was originally intended for managing network devices. However, numerous non-network systems have adopted SNMP to transfer log messages.
- Windows Event Log: Microsoft's proprietary logging format.
- Database: storing and retrieving log messages in a structured way.

The basic content for a log message is Timestamp + Source + Data. Regardless of the source of log message or how it is sent, these basic elements are always part of it. The timestamp marks the time at which the log message was generated. The source is the system that generated the log message, typically indicated in IP address or fully qualified hostname. Finally, the data is the description of an occurring event. There is unfortunately no standard format for how data is represented in a log message.

Log analysis is the process of analyzing log data to derive meaning from it. Having your data in a single place, the loghost, allows you to pull together or correlate log messages, which is especially important in a highly distributed environment in which multiple remote log collectors are implemented.

For each team taking part of the logging process, there is a specific log management need. Security teams are often solely interested in Security information and event management (SIEM) related events; the IT team is usually looking for problem resolution clues to derive from log data; the audit team may be interested in the usage of sensitive data. Other insights to derive from log data involve forensics, managing identity in cloud [21], marketing and accounting.

B. Managing Logs Generated from the Cloud

The process of analyzing logs from different sources plays a crucial role in digital forensic investigation. Virtual machine logs, network logs, and software logs together are really useful to perform a number security and auditing tasks. However, collecting this important information in the cloud environment is not as simple as it is in privately owned computer system, sometimes even impossible.

Logging in the Cloud is substantially different than traditional and distributed system logging; virtual machines and services have life cycles and their logs only survive the duration of the instance. As a consequence, it is very difficult to collect and prove the validity of the logs. For example, there's no way a can forensic investigator collect log files of malicious VMs, which have been terminated by an attacker after launching a DDoS attack [10]. This challenging aspect of making sense of log data generated by the cloud served as the premise for a previous work of ours [11] which argues for the validity of deploying logging standards in the cloud and presents a solution design to allow correlation and the ability to trace activities occurring within the cloud in a standard fashion. This present work, although self-sufficient, is, in many ways, a logical continuation from the ideas presented in our previous effort and has the same drive to improve logging conditions in the cloud.

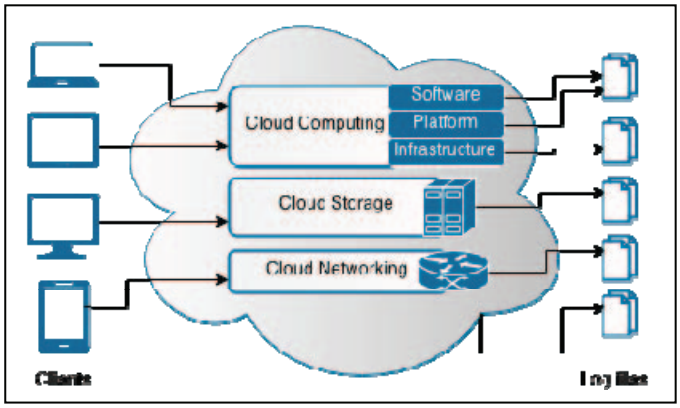

Figure 1. High level description of log data sources

Generally, Individual log messages should, at the very least, contain the totality of information to answer the question: "who performed which action and when?" Figure 1 gives a high level description of the different sources of log data across the cloud infrastructure. Cloud computing has the power to drive the rapid growth of economies and technological leaps, as well as providing organizations the ability to focus on their core business without concerning themselves with infrastructure, flexibility, and availability of resources [12]. The cloud ecosystem consists of the major IT cornerstones allowing to perform the tasks of computing (service models

typically consist of PaaS, SaaS, and IaaS), Storage (many distributed, highly fault tolerant, resources acting as one - often referred to as federated storage clouds [13]), and Networking (referring to the high-performance interconnected computer resources, communicating through high-bandwidth and low-latency specialized networks [24]).

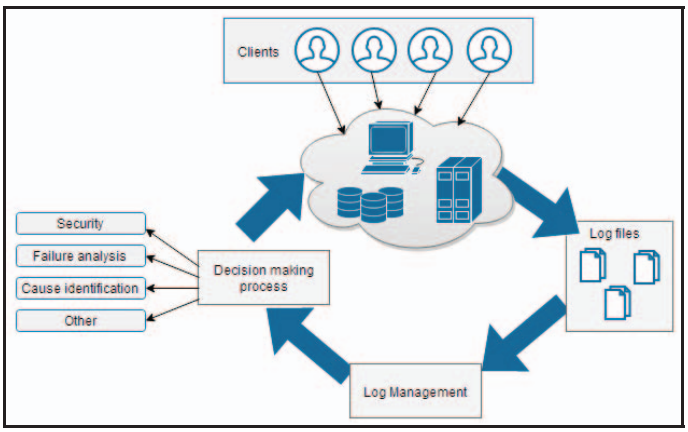

Figure 2. Cyclic nature of log management

In the cloud framework, we need application, system, and security logs from all software components. Next, we will want logs from the hardware infrastructure, virtual abstraction, routers, switches, Firewalls, etc. meaning that all infrastructure components in the cloud application stack is contributing to the ever growing size of log data, in order to achieve maximum correlation and ability to produce new insights during the log management process, which, in turn, helps shaping strategic goals and enterprise-level decision making (for example, security performance, causes to problems, alternative actions, etc.). Figure 2 offers a look into the cyclic nature of the log generation, management process; which allows obtaining insight from each time the natural movement of that cycle is taking place, allowing for the mitigation of cloud computing vulnerabilities, threats and ricks [22].

Some the most challenging aspects of cloud logging are:

- Decentralization: In the cloud, log data is not residing at one single centralized log server; rather, logs are decentralized across several servers. There are several layers and tiers in cloud architecture, and logs are generated in each tier.
- Size: as the volume and variety of log files rises immensely, it becomes increasingly difficult for log management tools to parse log files and trace potential issues; especially, when cross-log correlations are taken into consideration.
- Accessibility: Being generated in different layers, the logs need to be accessible to different actors of the system. For example, Security administrators need SIEM related events; the IT teams need relevant logs for troubleshooting, etc.
- Multi-tenancy: In cloud computing, multiple virtual machines (VM) can share the same physical infrastructure, which means that logs belonging to certain client might be mingled with another client logs. In a previous effort, we have proposed a solution design to this very problem [11]
- Unstructured log files: Log data is a prime example of unstructured information that is enclosed in fields that does not have a pre-defined data model. Log records are text-heavy and include also data such as dates and host addresses. Many organizations believe that unstructured data contains information that will help make better decision. However, it is very difficult to analyze this type of data.

As a consequence of the previously mentioned challenges, actual log management tools cannot efficiently manage the cloud log data. Therefore, a novel approach is needed.

III. Big Data Technologies to Treat Cloud Log Data

Stephen Marsland said in his book, "if data had mass, the earth would be a black hole" [14]. This is essentially what led the Big Data movement that takes in, manages, and analyzes massive volumes of data. It has been calculated that a company that has over 5000 servers, 2000 routers and switches, approximately 600 security devices (Firewalls, IDS, IPS, VPN, content service switches, Anti-Virus), and 50,000 plus desktop firewalls and host anti-virus (desktops and laptops), generates about 100,000 log messages per second from all systems [8]. Assuming that a log message is 300 bytes in average, quick math calculations results in 28.6 MB/s, 100.6 GB/h, 2.35 TB/day, and 860.5 TB/year. The sheer volume of this unstructured data is enormous.

As of 2012, it has been estimated that about 2.5 exabytes were created each day, and that number is doubling every 40 months or so [16]. An exabyte is one billion gigabytes, which offers organizations the opportunity to work with massive amounts of data. This rise in volume marks one of the three important attributes that gave rise to the big data movement [25]. Velocity, as another driving attribute of big data, signifies the speed of data creation in real time or near real time mode, which makes it possible for an entity to have a much quicker respond to its needs or have an upper hand over a competitor. Lastly, Variety indicates the many different sources that big data takes, an important number of which is relatively new. The data available is often unstructured, but there is a great deal of meaning to be derived from it when exploited with the proper tools. Machine-generated log data in the cloud checks all of the attributes that drive big data, which makes it only natural to treat it as such.

In traditional logging guidelines and log management, it is considered a best practice to not log anything that is of little value. It is a good principle, seeing as traditional relational databases (RDBMS) are made to deal with data-intensive applications and stringent queries. In recent years, another database system has been developed to serve a purpose different to that of RDBMS; which is NoSQL.

NoSQL is "Next Generation Databases mostly addressing some of the points: being non-relational, distributed, open-source, and horizontally scalable. The original intention has been modern Web-scale databases. The movement began early

2009 and is growing rapidly. Often more characteristics apply as: schema-free, easy replication support, simple API, eventually consistent /BASE (not ACID), a huge data amount, and more [15]."

The core of the big data movement, which fulfills its requirements in storing and processing the data in a fault tolerant, parallel and scalable fashion, is Hadoop. Hadoop [18] is an Apache open source software framework for reliable, scalable, distributed computing of massive amount of data. It was first developed and released by Yahoo!, it implements the MapReduce approach pioneered by Google. The MapReduce [26] framework constitutes the way Hadoop understands and assigns work to multiple servers operating on the data, which a "map" stage and a "reduce" stage. To store data, Hadoop utilizes its own distributed file system, HDFS [27], which distributes the data across multiple computer nodes, using abstraction of data replication, automatic handling of node failures [17]. Hadoop is also supplemented by an ecosystem of open source projects, including HBase [31], a NoSQL data store, and others like Hive [28], Pig [29] and Avro [30].

IV. PROPOSED METHODOLOGY

Machine-generated log data's extreme volumes, varieties and velocities can overwhelm unless it is filtered with great efficiency and rigor. In this section, we first give an overview of the proposed methodology in Figure 3, followed by descriptions of the different steps.

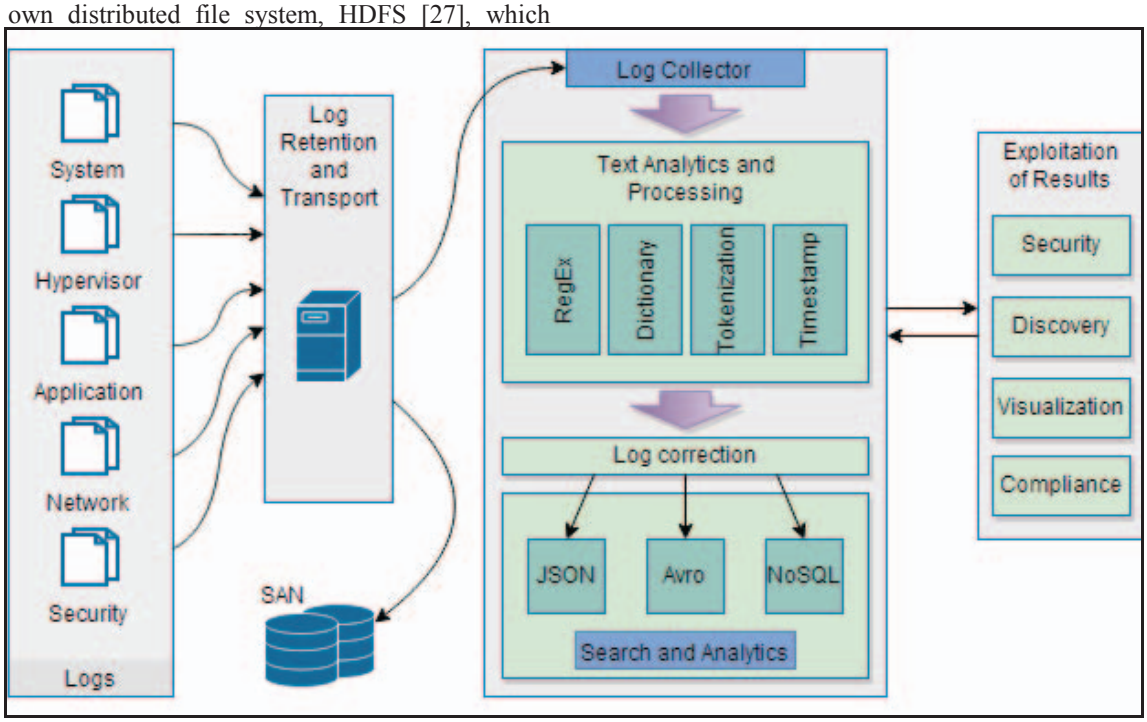

Figure 3. Architecture of our proposed methodology

A. Log Retention and Transport

Each cloud component that is generating log data must establish, maintain and make available, for the Log Retention and Transport layer, a systematic process of recording and retention of the log files. In traditional and distributed systems, the default log retention time is around 100 days, as well as taking into account fixed and rigid retention policies that rely on file count and size limit. Using Big Data distributed file systems in the cloud, it is recommended to increase the retention capabilities in all aspects, in order to extend and diversify the available log data. Log retention is done by running a script against the cloud platform stack to locate and identify log files, either scanning system-wide or statically searching in a given directory. This layer will be responsible for transporting the data to Storage Area Networks (SAN) for offboard storage, as well as to the next layer which is responsible for processing and parsing the data.

B. Log Collector

This layer serves the purpose of setting up a centralized solution so that multiple logs can be aggregated in a central location. This is due to the difficulty of managing and accessing hundreds of log files from hundreds of servers without using the proper tools. One of the traditional log management challenges is simplifying the approach to set up file replication on a central server. In this proposed methodology, the log collector stores all the data in HDFS across a number of computer clusters, allowing or high-volume and high-throughput log and event collection. The idea is to have the collection layer be horizontally scalable in order to increase in size with the increase number of log files and messages.

C. Text Analytics and Filters

Most of the data in the world is unstructured or semi-structured text. Log data is a prime example of unstructured useful information that is confined in description fields, and sometimes even discarded. Text analytics refers to the practice of deriving high-quality information from text and make available for further processing and analysis. As a consequence, using Text analytics techniques on log data will result in much more vigorous understanding of the knowledge residing within.

Essentially, the process of text analytics is writing rules to extract information from unstructured data. The way this level of the system works is writing those rules to a number of well defined filters. We have identified 4 very useful filters, but this list is not exhaustive in any way, which allows room for more filters to be included as plug-ins to the system. The regular expressions filter is perceived to recognized patterns of text, which allows us to extract information from the log message like a file name, web address, hostname, etc. the dictionary (or annotation) filter's purpose is locating a labeled text that matches a particular criteria, using a list of entries that contains domain specific terms (for example, a dictionary with a list of IT operations or companies stored as .dict file). The tokenization filter detects token boundaries, which is helpful to solve the problem of reading a log message that is separated in multiple lines. A good idea would be to merge the multiple lines into one line, testing if the line is starting with a whitespace. The timestamp filter is dedicated to correlate between the different timestamp formats that exist today, which is quite problematic if there is no solution set up for it.

D. Log Correction

The primary purpose for this layer is to modify incoherent log messages inside a file. An example for this is negative request duration in an apache log file. Apache, as well as other logging platforms, uses the gettimeofday() function. However, Network Time Protocol (NTP) skew affects it in order to fix the clock and account for daylight savings. In this case, time skew backwards gives rise to a negative request time. This layer has the ability to modify such information inside the log message.

E. Outputs, Search and Analytics

This layer is responsible for generating and indexing the previously extracted log data through a number of outputs. JSON (Javascript Option Notation) is great choice for the output, because it is a lightweight and open source format that transmits data objects as key-value pairs. This makes it easy to search by field, especially when using Jaql [32] (a JSON query Language) against it. Using Jaql enables modular query processing for semi-structured data and transparently exploiting massive parallelism through Hadoop's MapReduce. Avro and NoSQL databases (like HBase) outputs also provide search techniques that are very valuable for the exploitation of results.

F. Example of a Use Case

As a work in progress, we give a simple use case example to illustrate our proposed solution. In this use case, we take a single Apache common log record and run an AQL statement against it. AQL is a rule language for text analytics with which we can define custom rules for splitting and annotating log records. AQL is similar to Structured Query Language (SQL) where data generated by executing statements is stored in tuples. A collection of tuples generated for a given statement forms a view which is the basic AQL data model [19].

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -
0700] "GET /apache_pb.gif HTTP/1.0" 200
2326

LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

Figure 4. An Apache Common Log Format entry

Figure 4 displays a log file entry produced in Common Log Format (CLF), which is an Apache standard format that can be produced by different web servers [20]. Different parts of the entry are described in the pattern below the entry, where: `%h` is the IP address of the client that made the request, `%l` is the hyphen in the output, `%u` is the userid as determined by HTTP authentication, `%t` is the timestamp, `\"%r\"` is the request line from the client containing the method, URL and protocol, `%>s` is the status code that the server sends back, and `%b` is the size of the object returned to the client.

Once we know the details and the structure of the log entry, we can run AQL statements against it to extract data such as shown in figure 5.

The code in figure 5 creates a view 'LogInfo' from the text containing the log entry using a regular expression filter. The regular expression itself is constructed to be able to parse the IP address, the method used by the client (in this case GET), the resource URL that the client requested, and finally, the protocol used in the request. The expression itself results in 4 captured groups containing the information mentioned before. The statement goes on to assign names to each captured group.

```
create view LogInfo as
extract regex
/^(\S+)\s(?:\S+\s+)+"(\S*)\s?((?:[^"]*(?:\\")?)*)\s([
^"]*)"\s|/
        on L.text
        return group 1 as ip
                and group 2 as method
                and group 3 as url
                and group 4 as protocol
        from Document L;

create dictionary UserNames as ('frank', 'john',
'lucy');

create view UserID as
extract dictionary 'UserNames'on L.text as user from
Document L;
```

Figure 5. AQL statements to extract data

Next, we create a dictionary 'UserNames' that will contain only 3 names for the sake of the example. After that, we create a view called 'UserID' by applying the created dictionary to the log document we want to analyze. Views are table like structures with columns, rows and data types underlying the AQL extraction. The keyword 'Document' is the one view already available in each AQL extraction. Its most important column is "TEXT" which contains the text representation of the currently analyzed document.

```
{
    "ip": "127.0.0.1",
    "method": "GET",
    "url": "/apache_pb.gif",
    "protocol": "HTTP/1.0",
    "user": "frank"
```

Figure 6. Example of the output JSON file

Using the results of the views created, we can create yet another view to combine all the info we are interested into one single view, which will be loaded to a JSON file for best performance and increased file loading speeds. Using JSON files for storing extracted data is beneficial since most programming languages and analytics solution tools have support for it. The output JSON file will look like figure 6.

V. CONCLUSION

Evaluating machine generated log files immensely helps when diagnosing issues, and storing those files is generally required for security and compliance reasons. However, the sheer number of files can be overwhelming, especially in the case of cloud providers. Log data is also famed for being unstructured. Therefore, it is crucial to the cloud to consider the wider advantages of log management and analysis, as well as to harness technologies and services that allow a more complete exploitation of log data in the future.

In this paper, we have surveyed the traditional log management limitations and the challenges of implementing them in a cloud computing environment. We have also identified the requirements of unlocking the valuable wealth of information residing in log data generated in the cloud. Subsequently, we have argued in favor of using big data technologies to achieve the numerous enterprise- level objectives. Lastly, we have proposed a new methodology to meet the requirements and challenges we have put forth, taking into account the different phases of log retention and transport, to applying text analytics filters and correction tools, to outputting the extracted data in structured format ready for analysis and exploitation.

We believe that reporting and correlation through logging and log management should be made easy to use and fit to meet current and future goals. This work has been a step towards meeting those objectives. As a work in progress, our next immediate step is to find the right cloud environment to implement our solution in a complete and fully automated fashion. Furthermore, we are looking to include the discipline of machine learning in our proposed solution, seeing as how heterogeneous log datasets are growing more complex and inscrutable, important and compelling data variables and relationships are not at all clear in advance of the analysis. In the other hand, Stream computing can be used to enable security functions in real time. In fact, we can analyze log files, using the proposed methodology, to detect abnormal behavior. For example, an unusual number of packets from the same IP address could mean that a potential attack is taking place. As a result, we can alert the concerned server to take action. Stream computing and machine learning can be used to improve security and prevent attacks before the damage has been completed.

REFERENCES

[1] Mell, Peter, and Tim Grance. "The NIST definition of cloud computing." (2011).

[2] Sosinsky, Barrie. Cloud computing bible. Vol. 762. John Wiley & Sons, 2010.

[3] Foster, Ian, et al. "Cloud computing and grid computing 360-degree compared." Grid Computing Environments Workshop, 2008. GCE'08. Ieee, 2008.

[4] "Global Cloud Computing Market Forecast 2015-2020." Market Research Media. N.p., 08 Jan. 2014. Web. 25 Feb. 2015. <http://www.marketresearchmedia.com/2012/01/08/global-cloud-computing-market/>.

[5] FBI annual report for fiscal year 2007. Regional Computer Forensics Laboratory Program, 2008.

[6] Marinescu, Dan C. Cloud computing: Theory and practice. Newnes, 2013.

[7] Shackleford, Dave. Virtualization Security: Protecting Virtualized Environments. John Wiley & Sons, 2012.

[8] Chuvakin, Anton, Kevin Schmidt, and Chris Phillips. Logging and log management: The authoritative guide to understanding the concepts surrounding logging and log management. Newnes, 2012.

[9] Kent, Karen, and Murugiah Souppaya. Guide to computer security log management: recommendations of the National Institute of Standards and Technology. US Department of Commerce, Technology Administration, National Institute of Standards and Technology, 2006.

[10] Zawoad, Shams, Amit Kumar Dutta, and Ragib Hasan. "SecLaaS: secure logging-as-a-service for cloud forensics." Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security. ACM, 2013.

[11] Lemoudden, M., N. Ben Bouazza, and B. El Ouahidi. "Towards achieving discernment and correlation in cloud logging". In press.

[12] Aceto, Giuseppe, et al. "Cloud monitoring: A survey." Computer Networks 57.9 (2013): 2093-2115.

[13] Vernik, Gil, et al. "Data on-boarding in federated storage clouds." Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on. IEEE, 2013.

[14] Marsland, Stephen. Machine learning: an algorithmic perspective. CRC press, 2014.

[15] "List Of NoSQL Databases [currently 150]." NOSQL Databases. N.p., n.d. Web. 25 Feb. 2015. <http://nosql-database.org/>.

[16] McAfee, Andrew, and Erik Brynjolfsson. "Big data: the management revolution." Harvard business review 90 (2012): 60-6.

[17] Monteith, J. Yates, John D. McGregor, and John E. Ingram. "Hadoop and its Evolving Ecosystem." IWSECO@ ICSOB. 2013.

[18] White, Tom. Hadoop: The definitive guide. " O'Reilly Media, Inc.", 2012.

[19] "Annotation Query Language (AQL) Reference." Annotation Query Language (AQL) Reference. N.p., n.d. Web. 25 Feb. 2015. <http://www-

01.ibm.com/support/knowledgecenter/SSPT3X_2.0.0/com.ibm.swg.im.infosphere.biginsights.text.doc/doc/biginsights_aqlref_con_aql-overview.html>.

[20] "Log Files." - Apache HTTP Server. N.p., n.d. Web. 25 Feb. 2015. <http://httpd.apache.org/docs/1.3/logs.html>.

[21] Naoufal Ben Bouazza, Mouad Lemoudden, and Bouabid El Ouahidi. "Surveing the challenges and requirements for identity in the cloud." Security Days (JNS4), Proceedings of the 4th Edition of National. IEEE, 2014.

[22] Lemoudden, M., et al. "A Survey of Cloud Computing Security Overview of Attack Vectors and Defense Mechanisms." Journal of Theoretical and Applied Information Technology 54.2 (2013).

[23] Hashem, Ibrahim Abaker Targio, et al. "The rise of "big data" on cloud computing: Review and open research issues." Information Systems 47 (2015): 98-115.

[24] DeCusatis, Carolyn J. Sher, and Aparicio Carranza. "Cloud Computing Data Center Networking." Handbook of Fiber Optic Data Communication: A Practical Guide to Optical Networking (2013): 365.

[25] Zikopoulos, Paul, et al. Harness the Power of Big Data The IBM Big Data Platform. McGraw Hill Professional, 2012.

[26] Dean, Jeffrey, and Sanjay Ghemawat. "MapReduce: simplified data processing on large clusters." Communications of the ACM 51.1 (2008): 107-113.

[27] Shvachko, Konstantin, et al. "The hadoop distributed file system." Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on. IEEE, 2010.

[28] Thusoo, Ashish, et al. "Hive: a warehousing solution over a map-reduce framework." Proceedings of the VLDB Endowment 2.2 (2009): 1626-1629.

[29] Olston, Christopher, et al. "Pig latin: a not-so-foreign language for data processing." Proceedings of the 2008 ACM SIGMOD international conference on Management of data. ACM, 2008.

[30] Hoffman, Steve. Apache Flume: Distributed Log Collection for Hadoop. Packt Publishing Ltd, 2013.

[31] George, Lars. HBase: the definitive guide. " O'Reilly Media, Inc.", 2011.

[32] Beyer, Kevin S., et al. "Jaql: A scripting language for large scale semistructured data analysis." Proceedings of VLDB Conference. 2011.