An Attentive Security Mechanism to OpenStack Cloud Environment Using Enhanced Log Analysis

Sreekanth D, Gladston Raj S

Abstract: ***The security threat is a significant problem to the cloud environment in the current era. Providing security to the cloud without compromising the unique features of cloud computing such as high availability, scalability, etc. are needing to address whenever propose a security solution to the cloud environment. The authentication mechanisms and deployment of various cryptanalytical algorithms are one of the effective ways to prevent data leakage. Even though the security mechanisms can provide security to the data environment, there are different performance issues which will severely affect the unique features of cloud computing. This work has implemented with a method of deploying security to the cloud environment using the analysis of various cloud system logs collected from the OpenStack cloud.***

Keywords: OpenStack, RSYSLog, Exploratory Analytics

1. INTRODUCTION

Cyber Forensic activities can perform in the cloud environment is a hurdle due to the deployment of cloud instances geographically at various locations. One of the best ways to manage this issue is running some preventive mechanisms in the back ground. The proposed model is capable to collect and process and provide an alert mechanism to the OpenStack[1] cloud environment without compromising the unique featured of cloud computing.

2. OPENSTACK CLOUD

OpenStack is a project suite can use as a software-defined package to create the cloud environment. This environment facilitates with a computing facility, network, and storage amenities together in a single platform and the earlier days, it was managed separately as various units[2]. The OpenStack administrator has the right to choose the desired features to be deployed for the customized usage. It is a universal truth that the applications are highly available depends upon the demand[3].

The OpenStack-Ansible project is a popular method can use for the deployment of OpenStack environment. It refers to the facility to administrators to deploy the OpenStack consistently[4]. System configuration and management are easily possible by using the model and had god a wide acceptance among system administrators. In this research, the OpenStack cloud environment has created using OpenStack-Ansible project.

3. OPENSTACK LOGS

Logs are essential components of every application to make the environment auditable to assure the continuous improvement in the performance of the system. An improved analysis of system logs can treat as a third eye of the system for constant improvements. There is a vital requirement for the deployment of an effective mechanism for the analysis of cloud logs gathered from various cloud instances. Continuous improvement on the performance of the system, anomaly detection through the of the system, etc. can be effectively deployed through the log analysis.

4. PROPOSED MODEL

The model deployed is explained using the diagram. The logs will be collected from various OpenStack locations using RSYSLog system. There are facilities to store the data as well as the data can forward to the exploratory analytics module. The assumption derived can save, or in the case of any anomalies, it can report to the authority.

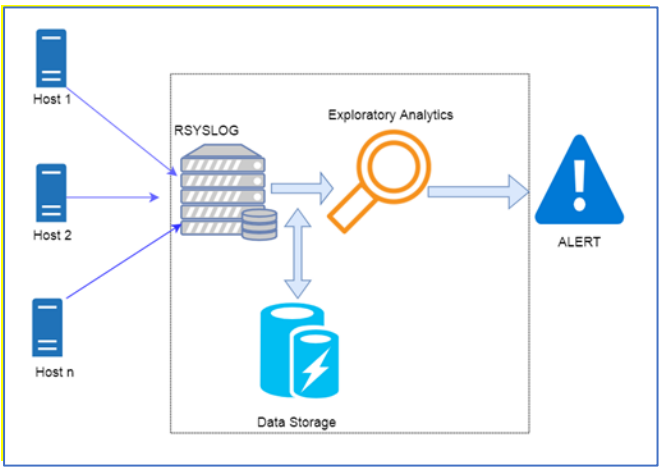

Figure 1 –Log Analysis Model

The model developed for effective log analysis is depicts in the figure 1

4.1 Introduction to Logs

Every transaction in the environment adequately logged traditionally[5]. Various logs generated at multiple events will store at numerous locations. The logs can take from the common subdirectory /vary/log/directory.

Revised Manuscript Received on April 15, 2019.

Sreekanth D, Research Scholar, Barathiar University, Coimbatore, Tamilnadu, India

Gladston Raj S, Head, Dept of Computer Science, Govt. College, Nedumangad, Trivandrum, Kerala, India

4.2 Type of Logs

OpenStack provides a standard logging mechanism of generating the logs by categorizing it at multiple levels as TRACE, AUDIT, INFO, WARNING, ERROR and CRITICAL. The words will present on the logs based on the type of activities taken place at the environment.

Every transaction takes place will be recorded appropriately as logs, and it is highly useful if an instance fails, or misbehavior happens in the environment[6].

The activity logs can be easily traced to know about the events happening at the system and can provide a security mechanism if it is malicious without making any compromises on the performance of the system[7]. The details of the events happening can be tracked using the UUID associated with it along with service logs.

The cloud environment is composed of several servers and needs to collect the logs separately from these places. It is always good to receive the logs to a central place to assure better analysis[8]. The central logging is a hard process because different operating systems will be functioning at various locations.

4.3 RSYSLOG

RSYSLOG is the fastest system for log handing out. It provides the facility to perform the quickest operation and also have higher security features in handling the log files. Security to the log files is also very much important because it is an opening to the access of every activity happening at hosted application. An intruder can plan the attack based on the continuous analysis of the logs collected. RSYSLog follows a modular design approach, and it can generate the fastest response and also will be helpful in a compartment thoughtful log analysis.

RSYSLog is the fastest system provides the facility to collect the records beyond one million per second to the central log locations. The system facilitates a remarkable speed even if it need to obtain the logs from various sources.

OpenStack – Ansible project supports to collect the logs from various sources through the RSYSLog, and the file will obtain in openstack_user_config.yml. One of the other big problems is why should keep the records of all logs? Handling this trillions of records itself is a trouble, it can't publicize because of the sensitive information contained. RSYSLog will compress the file within a specific period and can reuse in the case of any particular auditing requires on the previous records.

The log records can collect from various locations in the OpenStack-Ansible project, and some of the available areas discussed here:

- The logs can gather from the location /var/log/log-storage within directories or physical host.
- The physical log contains its service containers also riding at /openstack/log/.
- The service containers itself maintains the logs at /var/log/<name of service>

Data collected through RSYSLog can use for exploratory analytics, and in the case, the data can also get stored in the databases like MySQL or MongoDB, etc. for further processing.

5. RESULTS AND DISCUSSIONS

Monitoring the logs is a difficult task because of the massive generation of logs at various sources. In every up and running system mainly on the cloud environment, the log monitoring became an individual requirement to track every action takes place at the network[9]. Apart from the traditional security settings deployed at the host level and application level, this can be treated as an essential component to run a crawling job at the history to track and generate alarms in the case of any anomalies detected.

Various type of log files is available, and the big question is what kind of approach can give an accurate result based on scientific analysis[10]. The logs can be collected a regular expression and need to parse it for making it meaningful. A correct log file generated at OpenStack will consist of the details of access to the aspects of the operation performed.

- 2018-06-20 16:42:10.567 38397 INFO nova.virt.libvirt.driver [-] [instance: b1b8e5c7-12f0-4092-84f6-297fe763245] Instance spawned successfully
- 2018-06-20 16:42:14.307 38397 INFO nova.virt.libvirt.driver [-] [instance: b1b8e5c7-12f0-4092-84f6-297fe763245] Instance destroyed successfully

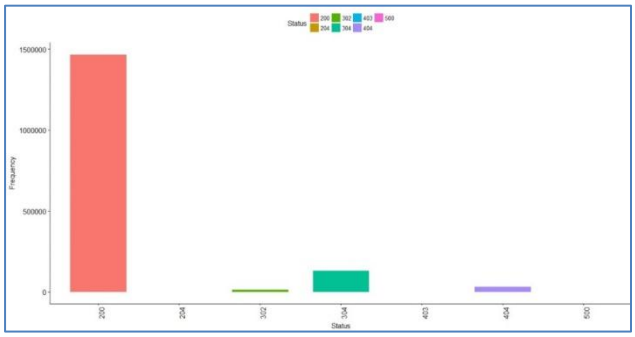

Figure 2: Status Codes

Figure 2 depicts the information derived after analyzing status codes. It can see that there is a demand for the outstanding service request and there are some other requests too. If that is the case, the analysis can make from the result is the anomalies can track by performing the inverse analysis such as on a smaller number of transactions[5].

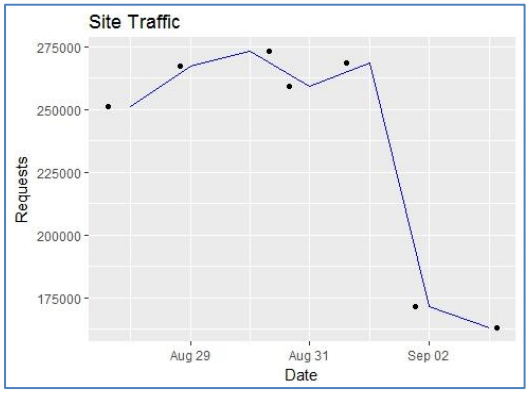

Figure 3: Site Traffic

Figure 3 depicts the traffic to the site reported during the period. By making exploratory analytics at logs, it is highly useful to assume on the peak time and also if there is a sudden dip in the usage in between[11].

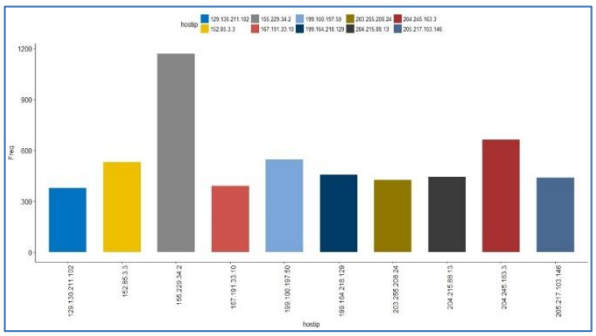

Figure 4: IP Based Website Hits

Figure 4 deploys the IP based Website Hits, and the IP requests can obtain in OpenStack by mapping the instance details. The grouping of requests based on the status and instances also can be derived, and this will be more helpful to tack the anomalies.

5.1 Alert Mechanism

The alert mechanism is a module deployed in this model based on the exploratory analytics. Based on the analysis made at various levels like correlating time stamp and usage consumption, status alerts and Instance details, different text log generated by the[12] system, etc. the system will detect the anomalies. Based on the systematic analysis of various layers it will identify the threats and will report to the customer. Based on the customer's feedback, the system is capable of deciding whether to block or allow access to the instance.

Process flow:

[1] Logs collect from various locations
[2] Perform Log Parsing
[3] Apply exploratory analytics
[4] Record the results obtained

Initiate alerts (if needed)

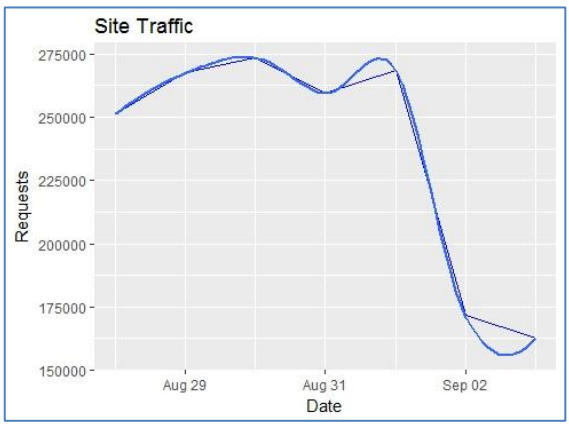

Figure 5: Site Traffic Trend

Based on the data, a trend on the usage can be derived, and any abnormalities happen to the system other than the identified direction the system can monitor. Any change occurs to the trend can be treated as a first level warning message towards the need of the system monitor as in the figure 5.

6. CONCLUSION

Here a model deployed for collecting the logs from various sources through an OpenStack Environment and developed an effective alert generating system through the exploratory analytics. The model implemented is capable of receiving the logs from multiple sources, and there is a facility to store the information on the analysis made. The exploratory analytics module will identify the threats by performing different analysis such as Status tracking, Instance Analysis, and Trend Mapping, etc. The system is capable of collecting feedback from users and will work based on the user's input.

REFERENCES

1. J. Wei, Y. Zhao, K. Jiang, R. Xie, and Y. Jin, "Analysis farm: A cloud-based scalable aggregation and query platform for network log analysis," *Proc. - 2011 Int. Conf. Cloud Serv. Comput. CSC 2011*, pp. 354–359, 2011.
2. V. Chang, Y. H. Kuo, and M. Ramachandran, "Cloud computing adoption framework: A security framework for business clouds," *Futur. Gener. Comput. Syst.*, vol. 57, pp. 24–41, 2016.
3. N. Paladi, C. Gehrmann, and A. Michalas, "Providing User Security Guarantees in Public Infrastructure Clouds," *IEEE Trans. Cloud Comput.*, vol. 5, no. 3, pp. 405–419, 2017.
4. S. Sabitha and M. S. Rajasree, "Access control based privacy preserving secure data sharing with hidden access policies in cloud," *J. Syst. Archit.*, vol. 75, pp. 50–58, 2017.
5. M. Farshchi, J. G. Schneider, I. Weber, and J. Grundy, "Experience report: Anomaly detection of cloud application operations using log and cloud metric correlation analysis," *2015 IEEE 26th Int. Symp. Softw. Reliab. Eng. ISSRE 2015*, pp. 24–34, 2016.
6. S. Muthurajkumar, S. Ganapathy, M. Vijayalakshmi, and A. Kannan, "Secured temporal log management techniques for cloud," *Procedia Comput. Sci.*, vol. 46, no. Icict 2014, pp. 589–595, 2015.
7. B. J. Jansen, "Search log analysis : What it is, what ' s been done, how to do it," vol. 28, pp. 407–432, 2006.
8. I. Mavridis and H. Karatza, "Performance evaluation of cloud-based log file analysis with Apache Hadoop and Apache Spark," *J. Syst. Softw.*, vol. 125, pp. 133–151, 2017.
9. W. C. Kuo, H. J. Wei, and J. C. Cheng, "An efficient and secure anonymous mobility network authentication scheme," *J. Inf. Secur. Appl.*, vol. 19, no. 1, pp. 18–24, 2014.
10. B. Alami Milani and N. Jafari Navimipour, "A comprehensive review of the data replication techniques in the cloud environments: Major trends and future directions," *J. Netw. Comput. Appl.*, vol. 64, pp. 229–238, 2016.
11. R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning For Network Intrusion Detection," *2010 IEEE Symp. Secur. Priv. Outs.*, pp. 305–316, 2010.
12. D. Holmes, "2016 DDoS ATTACK TRENDS," *Underst. DDoS Attack*, no. August, 2016.