Journal of Science and Technology (JST)
Volume 2, Issue 5, Sep - October 2017, PP 42-46
www.jst.org.in *ISSN: 2456 - 5660*

Protecting Virtualized Infrastructures in Cloud Computing Based On Big Data Security Analytics

Deshpande Chandrika[1], Dr. M. Sreedhar Reddy[2]
[1]*(Department of CSE Samskruti College of Engineering and TechnologyKondapur, Ghatkesar,Hyderabad)*
[1]*(Department of CSE Samskruti College of Engineering and TechnologyKondapur, Ghatkesar,Hyderabad)*

Abstract : *Virtualized infrastructure in cloud computing has become an attractive target for cyber attackers to launch advanced attacks. This paper proposes a novel big data based security analytics approach to detecting advanced attacks in virtualized infrastructures. Network logs as well as user application logs collected periodically from the guest virtual machines (VMs) are stored in the Hadoop Distributed File System (HDFS). Then, extraction of attack features is performed through graph-based event correlation and MapReduce parser based identification of potential attack paths. Next, determination of attack presence is performed through two-step machine learning, namley logistic regression is applied to calculate attack's conditional probabilities with respect to the attributes, andbelief propagation is applied to calculate the belief in existence of an attack based on them. Experiments are conducted to evaluate the proposed approach using well-known malware as well as in comparison with existing security techniques for virtualized infrastructure. The results show that our proposed approach is effective in detecting attacks with minimal performance overhead.*

Keywords – *HDFS, VM*

I. INTRODUCTION

The three data driven platforms which have evolved since the advancement and origin of the Internet are Cloud Computing, Big Data and Virtualization. They continue to dominate the control, flow, and preservation of the data for many large scale and various other enterprises.Cloud Computing or („the Cloud‟) is a term that describes the collaboration, agility, scaling and availability. It provides for the cost reduction carried out through optimized and efficient computing [1]. The Cloud Computing environment is classified based upon its essential characteristics, three services models, and four deployment models, by the National Institute of Standards and Technology (NIST) [2]. The functionalities carried out by the Cloud, associated with an enterprise make it more vulnerable to attacks and breach in its security and privacy, thus making it an important asset to be protected.Big Data , as the name suggests is large amounts of data collected, processed and stored. The Big Data is classified based upon the four V‟s abbreviation. The four V‟s are: Volume, Velocity, Variety and Veracity [3]. Big Data analytics contain metadata and can be used to expose the privacy of an individual or any organization, security measures for the same need to be constructed.Virtualization infrastructure and the technology associated with it is developing a fast recognition in the industry. Virtualization is the process in which the virtual machine is created, and it generates a dual operating system environment setup on a single operating system. Fedora [4] is an example of a distro of the Linux operating system which can be installed on a virtual machine platform like VMWare [5] or Oracle Virtual Box [6] , and thus the Linux based operating system can be utilized on a Microsoft Windows running machine. The main idea of this paper is to put a limelight on the present security measures available to these information processing platforms and to put forth some ideas which can be implemented to provide additional protection and security to this data in order to maintain the consistency and integrity of the data.

II. S ECURITY OF THE CLOUD

The National Institute of Standards and Technology (NIST) defines Cloud Computing as, "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"[7].

A. Cloud Platforms The Cloud Computing platform can be further categorized by the services it offers and the models on which it can be deployed [8], [9]

- Infrastructure as a Service (IaaS) This is the platform that provides virtualized resources, storage and network connectivity to the users. Users are eligible to scale these resources on demand. IaaS maybe used as a high level cloud system. Common examples of this services are, Amazon EC2, GoGrid, Microsoft Azure.
- Platform as a Service (PaaS) It is the platform that provides development and virtualized resources which contain enhanced programming platform elements. It also supports geographically distribution of developers which contribute towards the development of the platform. Amazon Map Reduce/Simple storage, Google App Engine and Heroku are some of the examples of PaaS.

• Software as a Service (SaaS) The most advanced platform of Cloud Computing, it provides an on-demand access to the services mostly located on the web browser or the computer network. The features are available more frequently and moreover no license has to be purchased. Due to its easy availability, the SaaS platform can be integrated easily with mashup applications. Google Maps, Salesforce.com are some prominent examples.

III.LITERATURE REVIEW

Malware detection in virtualised infrastructure

Malware refers to any executable which is designed to compromise the integrity of the system on which it is run. There are two prominent approaches to malware detection in cloud computing, namely in-VM and outside-VM interworking approach and hypervisor-assisted malware detection.

In-VM and outside-VM interworking approach to malware detection

In-VM and outside-VM interworking detection consists of an in-VM agent running within the guest VM, and a remotescrutiny server monitoring the VM"s behaviour. When a potential malware execution is detected the in-VM agentsends the suspicious executable to the scrutiny server, which then uses the signature database to verify malware presence or otherwise and then informs the in-VM agent of the results. CloudAV, a cloud-based malware detection system featuring multiple antivirus engines, employs in-VM and outside-VM interworking approach to protect the guest VMs against attacks [4]. Apparently the effectiveness of this scheme depends on the frequency at which the virus signatures are updated by the antivirus vendors. The in-VM and outside-VM interworking approach is also used by CuckooDroid, to detect mobile malware presence on Android devices [5]. It consists of an in-device agent which scans executables on the device and sends any suspicious executable to a remote scrunity server which runs a hybrid of anomaly-based and signature-based malware detectors. The scheme first extracts malware features by using static as well as dynamic analysis on malware apps. The obtained features are then used to train a one-class SVM (Support Vector Machine) classifier for anomaly-based detection. Implemented on an emulated Android platform, CuckooDroid achieved a detection accuracy of 98.84 %.

Hypervisor-assisted malware detection

Hypervisor-assisted malware detection, on the other hand, uses the underlying hypervisor to detect malware within

the guest VMs. A hypervisor-assisted malware detection scheme is designed in [6] to detect botnet activity within the guest VMs. The scheme installs a network sniffer on the hypervisor to monitor external traffic as well as inter-VM traffic. Implemented on Xen, it is able to detect the presence of the Zeus botnet on the guest VMs. A hypervisor-assisted detection scheme is proposed in [7] using guest application and network flow characteristics. This scheme first uses LibVMI to extract key process features from the processes running within VMs and then uses tcpdump together with the CoralReef network packet analysis tool from CAIDA (Center for Applied Internet Data Analysis) to extract network flow features. The obtained features are then used to train one-class SVM classifiers to detect malware presence within guest VMs. Implemented on KVM, the scheme is able to detect well-known DDoS (Distributed Denial of Service) and botnet attacks such as LOIC (Low Orbit Ion Cannon) and Zeus. The hypervisor-assisted detection is also used in Access- Miner [8]. Implemented as a custom hypervisor, Access- Miner monitors normal user behavior within the system and creates access activity models which are used for anomalybased malware detection. To ensure that the underlying hardware is protected, it intercepts the guest system call requests and uses a policy checker module to determine if it should access the system resource.

Security Analytics

Security Analytics refers to the application of analytics in the context of cybersecurity [9]. Based on a variety of datacollected from different points within an enterprise network, security analytics aims to detect previously undiscovered threats by use of analytic techniques. Common techniques of security analytics include clustering and graph-based event correlation.

Clustering for security analytics

Clustering organises data items in an unlabeled dataset into groups based on their feature similarities [10]. For security analytics, clustering finds a pattern which generalises the characteristics of data items, ensuring that it is well generalized to detect unknown attacks. Examples of clusterbased classifiers include K-means clustering and k-nearest neighbors, which are used in both intrusion detection and malware detection. Clustering is used for security analytics for industrial control systems [11] in an NCI (networked critical infrastructure) environment. First, data outputs from various network sensors are arranged as vectors and K-means clustering is applied to group the vectors into clusters. The MapReduce model is then applied to the grouped clusters to find groupings of possible attack behaviour, thus allowing the detection to be carried out efficiently. In [12] an "attack pyramid" -based scheme is proposed to detect APTs (advanced persistent threats) in a large enterprise network environment. Based on threat tree modeling, different planes (namely hardware, user, network, application) to which an attack may be launched are placed hierarchically with the end goal placed at the top. First, outputs from all available sensors in the network (e.g., network logs, execution traces, etc) are put into contexts. Then, in terms of the contexts various suspicious activities detected at each attack plane are correlated in a MapReduce model, which takes in all the sensor outputs and generates an event set describing potential APTs. Finally, an alert system determines attack presence by calculating the confidence levels of each correlated event. SINBAPT (Security Intelligence techNology for Blocking APT) [13] uses big data processing such as HDFS and MapReduce together to detect the presence of APTs in an enterprise network environment. Used for anomaly-based detection, the scheme collects log data from different sources (e.g., Netflow, application logs, etc) and applies a MapReduce model for feature extraction. Once organized into clusters, the data is then used to determine attack presence according to pre-defined rules.

Graph-based event correlation

While clustering determines attack presence through grouping common attack characteristics, it is limited in establishing an accurate correlation which may exist between events. This makes it difficult to accurately identify the sequences of events leading to the presence of an attack within the network, as well as the entry point of the attack. Graph-based event correlation overcomes this limitation by representing the events from the logs obtained as sequences in a graph. Given a collection of logs from different points within the network (e.g., firewall logs, web server logs, etc.), these events are correlated in a graph with the event features (e.g., timestamp, source and destination IP, etc.) represented as vertices and their correlations as edges. This enables the accurate identification of the entry point which an attack enters, as well as the sequences of events which the attack undertakes. Graphs-based event correlation is used in BotCloud, a botnet detection system for large enterprise environments [14]. Based on the Netflow data which describe the various network traffic flows between clients, the scheme represents the network flow between clients in the form of a dependency graph. The graph is then input into a MapReduce model to identify network IP associations using PageRank algorithm. Graphs-based event correlation is presented in the security framework designed to detect attacks within critical infrastructures [15]. The scheme collects events from different sources within the network, and generates a temporal graph model to derive different event correlations for threat detection.

IV. PROPOSED APPROACH

3.1 Overall Framework

The basic idea of our proposed approach is to detect in realtime any malware and rookit attacks via holistic efficientuse of all possible information obtained from the virtualized infrastructure, e.g., various network and user application logs. Our proposed approach is a big data problem for the following characteristics of the network and user application logs collected from a virtualized infrastructure: _ Volume: Depending on the number of guest VMs and the size of the network, the amount of the network and user application logs to be collected can range from approximately 500 MB to 1 GB an hour; _ Velocity: The network and user application logs are collected in real-time, in order to detect the presence of malware and rootkit attacks, accordingly the collected data containing its behavior needs to be processed as soon as possible;

_ Veracity: Due to the “low and slow” approach that malware and rootkit take in hiding their presence within the guest VMs, data analysis has to rely upon event correlation and advanced analytics. The design principles, which are integral in the development of our BDSA approach to protecting virtualized infrastructures, can be elaborated as follows.

_ Design Principle # 1 - Unsupervised classification: The attack detection system should be able to classify potential attack presence based on the data collected from the virtualized infrastructure over time.

Design Principle # 2 - Holistic prediction: The attack detection system should be able to identify potential attacks by correlating events on the data collected from multiple sources in the virtualized infrastructure.

_ Design Principle # 3 - Real-time: The attack detection system should be able to ascertain attack presence as immediately as possible so as for the appropriate countermeasures to be taken immediately.

_ Design Principle # 4 - Efficiency: The attack detection system should be able to detect attack presence at a high computational efficiency, i.e., with as little performance overhead as possible.

_ Design Principle # 5 - Deployability: The attack detection system should be readily deployable in production environment with minimal change required to common production environments.

Figure 1 illustrates the overall conceptual framework of our proposed big data based security analytics (BDSA) approach, with the different components highlighted in blue. Our BDSA approach consists of two main phases, namely

_ Extraction of attack features through graph-based event correlation and MapReduce parser based identification of potential attack paths, and Determination of attack presence via two-step machine learning, namely logistic regression and belief propagation.

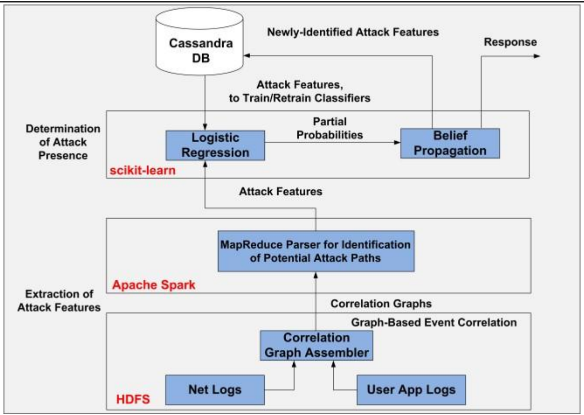
Cassandra DB
Newly-Identified Attack Features
Response
Attack Features,
to Train/Retrain Classifiers
Determination of Attack Presence
Logistic Regression
Partial Probabilities
Belief Propagation
scikit-learn
Attack Features
MapReduce Parser for Identification of Potential Attack Paths
Apache Spark
Extraction of Attack Features
Correlation Graphs
Graph-Based Event Correlation
Correlation Graph Assembler
HDFS
Net Logs
User App Logs

Fig. 1: Conceptual framework of the proposed big data based security analytics (BDSA) approach

Prior to the online detection of attacks, there is actually a system initialization, in which offline training of the logistic regression classifiers is carried out, that is, the stored features are loaded from the Cassandra database to train the logistic regression classifiers. Specifically, wellknown malicious as well as benign port numbers are loaded to train a logistic regression classifier to determine if the incoming/outgoing connections are indicative of an attack presence. Likewise, well-known malware and legitimate applications together with their associated ports are loaded to train a logistic regression classifier to determine if the behavior of an application running within the guest VM is indicative of an attack presence. These trained logistic regression classifiers are ready for online use, upon the extraction of new attack features, to determine if the potential attack paths are indicative of attack presence.

In the Extraction of Attack Features phase, first, it carries out Graph-Based Event Correlation. Periodically collected from the guest VMs, network and user application logs are stored in the HDFS. By assembling the information contained in these two logs, the Correlation Graph Assembler (CGA) forms correlation graphs.Then, it carries out the Identification of Potential Attack Paths. A MapReduce model is used to parse the correlation graphs and identify the potential attack paths i.e., the most frequently occurring graph paths in terms of the guest VMs" IP addresses. This is based on the observation that a compromised guest VM tends to generate more traffic flows as it tries to establish communication with an attacker. In the Determination of Attack Presence phase, two-step machine learning is employed, namely logistic regression and belief propagation are used. While the former is used to calculate attack"s conditional probabilities with respect to individual attributes, the latter is used to calculate the belief of an attack presence given these conditional attributes. From the potential attack paths, the monitored features are sorted out and passed into their logistic regression classifiers to calculate attack"s conditional probabilities with respect to individual attributes. The conditional probabilities with respect to individual attributes are passed into belief propagation to calculate the belief of attack presence. Once attack presence is ascertained, the administrator is alarmed of the attack. Furthermore, the Cassandra database is updated with the newly-identified attack features versus the class ascertained (i.e., attack or benign), which are then used to retrain the logistic regression classifiers.

V. CONCLUSION

In this paper, we have put forward a novel big data based security analytics (BDSA) approach to protecting virtualized infrastructures in cloud computing against advanced attacks. Our BDSA approach constitutes a three phase framework for detecting advanced attacks in real-time. First, the guest VMs"s network logs as well as user application logs are periodically collected from the guest VMs and stored in the HDFS. Then, attack features are extracted through correlation graph and MapReduce parser. Finally, two-step machine learning is utilized to ascertain attack presence. Logistic regression is applied to calculate attack"s conditional probabilities with respect to individual attributes. Furthermore, belief propagation is applied to calculate the overall belief of an attack presence. From the second phase to the third, the extraction of attack features is further strengthened towards the determination of attack presence by the two-step machine learning. The use of logistic regression enables the fast calculation of attack"s conditional probabilities. More importantly, logistic regression also enables the retraining of the individual logistic regression classifiers using the new attack features TABLE 6: Comparison of security approaches

REFERENCES

[1] D. Fisher, “““venom" flaw in virtualization software could lead to vm escapes, data theft,” https://threatpost.com/venomflaw- in-virtualization-software-could-lead-to-vm-escapes-datatheft/ 112772/, 2015, accessed: 2015-05-20.

[2] Z. Durumeric, J. Kasten, D. Adrian, J. A. Halderman, M. Bailey, F. Li, N.Weaver, J. Amann, J. Beekman, M. Payer et al., “The matter of heartbleed,” in Proceedings of the 2014 Conference on Internet Measurement Conference. Vancouver, BC, Canada: ACM, 2014, pp. 475–488.

[3] K. Cabaj, K. Grochowski, and P. Gawkowski, “Practical problems of internet threats analyses,” in Theory and Engineering of Complex Systems and Dependability. Springer, 2015, pp. 87–96.

[4] J. Oberheide, E. Cooke, and F. Jahanian, “Cloudav: N-version antivirus in the network cloud.” in USENIX Security Symposium, San Jose, California, USA, 2008, pp. 91–106.

[5] X. Wang, Y. Yang, and Y. Zeng, “Accurate mobile malware detection and classification in the cloud,” SpringerPlus, vol. 4, no. 1, pp. 1–23, 2015.

[6] P. K. Chouhan, M. Hagan, G. McWilliams, and S. Sezer, “Network based malware detection within virtualised environments,” in Euro-Par 2014: Parallel Processing Workshops. Porto, Portugal: Springer, 2014, pp. 335–346.

[7] M. Watson, A. Marnerides, A. Mauthe, D. Hutchison et al., “Malware detection in cloud computing infrastructures,” IEEE Transactions on Dependable and Secure Computing, pp. 192 –205, 2015.

[8] A. Fattori, A. Lanzi, D. Balzarotti, and E. Kirda, “Hypervisorbased malware protection with accessminer,” Computers & Security, vol. 52, pp. 33–50, 2015.

[9] T. Mahmood and U. Afzal, “Security analytics: big data analytics for cybersecurity: a review of trends, techniques and tools,” in Information assurance (ncia), 2013 2nd national conference on. Rawalpindi, Pakistan: IEEE, 2013, pp. 129–134.

[10] C.-T. Lu, A. P. Boedihardjo, and P. Manalwar, “Exploiting efficient data mining techniques to enhance intrusion detection systems,” in Information Reuse and Integration, Conf, 2005. IRI-2005 IEEE International Conference on. Las Vegas, Nevada, USA: IEEE, 2005, pp. 512–517.

[11] I. Kiss, B. Genge, P. Haller, and G. Sebestyen, “Data clusteringbased anomaly detection in industrial control systems,” in Intelligent Computer Communication and Processing (ICCP), 2014 IEEE International Conference on. Cluj-Napoca, Romania: IEEE, 2014, pp. 275–281.

[12] P. Giura and W. Wang, “Using large scale distributed computing to unveil advanced persistent threats,” Science J, vol. 1, no. 3, pp. 93–105, 2012.