Australian Journal of Basic and Applied Sciences, 9(33) October 2015, Pages: 51-55

ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

Time Sensitive Data Stream to Achieve Intrusion Prevention in IaaS Cloud

M. Suresh and C. Kavitha

Research Scholar, Anna University India, Computer Science, Thiruvalluvar Govt Arts College India

ARTICLE INFO

Article history:
Received 3 October 2015
Accepted 31 October 2015

Keywords:
Cloud Computing, Intrusion Prevention, Data Stream, Frequent pattern mining

ABSTRACT

Cloud computing gains its rapid traction in the commercial marketplace because of its ability to reduce the cost of IT Infrastructure, increased flexibility and scalability of computer processing. Though we have many pros there are few critical issues such as storage location, security and process outsourcing, etc., still exists. Out of this security of the primary issues which reduces the growth as well as the complication with data protection and data privacy continue to infection the market. Intrusion predictions in cloud paradigm are for high gains, may it be a awful user or a opponent of cloud client. Confronting new implementation scenario, traditional Intrusion Prevention Systems (IPS) are not well suited for cloud environment. To handle large scale network access traffic and managerial control of data and application in cloud, a new multi-threaded distributed cloud IPS model has been proposed. An Intrusion prevention system have been built with online mining of frequent item sets over a stream with Time-sensitive sliding window, which is one of the most vital technique in stream data mining with broad applications. This approach will be used to set the rules for backtracking as if needed to determine the intrusion characteristics then to implement the "Deny All except allowed" policy rules for prevention.

To Cite This Article: M. Suresh and C. Kavitha., Time Sensitive Data Stream to Achieve Intrusion Prevention in IaaS Cloud. ***Aust. J. Basic & Appl. Sci.,*** *9(33): 51-55, 2015*

INTRODUCTION

In the recent years, the viability of cloud computing has been heightened by many technologies such as Virtualization, Grid computing, Service Oriented Architecture and Hypervisor, lately this completely monitors Elasticity, Scalability and On-Demand. One of the main aspects is that the delivery model in IT ecosystem here is defined with anytime access to services, billed upon usage provided an internet connection. This delivery model lets Consumers and businesses access their personal files and applications without installation. Cloud computing serves an effective alternative for business with increasing computational resources mainly hardware and software which furnishes IT services built upon usage payment. Cloud computing is necessarily becoming stronger enabling new features and reducing costs in infrastructure and maintenance. However, the security issues are still it's vital anxiety as many security breach events have occurred. Cloud computing poses as a global integration of logic, data and users but still the major concern about security intercepts the enterprises for not availing cloud-based solutions. Security in cloud must remain a significant objective in cloud-based systems covering end users and application providers positively bidirectional. Security-aware software services are required by end users which inhibits their private data from being disclosed to cloud providers or other users. A methodology for security protected testing is required by the application providers to keep from leakage of internal activity and product features of the company to external sources (Subashini, S., V. Kavitha, 2011). Cloud computing can be deployed in the following service models commonly referred, Infrastructure-as-a-Service (IaaS) – It delivers network resources, storage and processing to host user systems and favours these resources through virtualization. To manage the application's workload, many servers, routers, storage systems and other equipment are made available. Platform-as-a-Service (PaaS) – It offers the users the facility to customize, manage , control and deploy their applications as it is an execution platform pertaining distribution and application development. This effortlessly manageable cloud application provides an applicative programming platform to a level of abstraction above Iaas. Software-as-a-Service (SaaS) – This upper cloud model provides access to applications owned by the service providers according to the users expectations and needs. On the basis of how resources are organized and made available to users

Corresponding Author: M. Suresh, Research Scholar, Anna University India, Computer Science, Thiruvalluvar Govt Arts College India
E-mail: sureshmadhesan@gmail.com

the following deployment strategies are classified. Public Cloud – Availability of resources to the general public which may also be owned and managed by academic, private, governmental and even their combinations. Private Cloud – Provisioning Physical resources exclusively to be used by a single organization. Community Cloud - A community of customers with similar specific needs, such as compliance, control and security serves and comprises the community cloud. Hybrid Cloud - Combinations of different infrastructures such as private, public or community and merging as a single unit of cloud, this infrastructure is set up.

Eucalyptus:

A system that majorly used across the globe to implement hybrid and private clouds is Eucalyptus (Daniel Nurmi, 2009). The architecture is quite simple and flexible with a modular hierarchical design that reflects common resource environments as in academic designs. This cloud computing research based system is interface-compatible with Amazon EC2 that enables to run without any modifications on both Amazon and Eucalyptus. The efficient Eucalyptus cloud computing platform enables flexible allocation techniques decoupled from specific hardware resources using the virtualization capabilities of the hypervisor. The Eucalyptus installation is accomplished through four high-level components with its independent web-service interfaces. Node Controller administers the inspection, execution and termination of VM instances on the host. Cluster Controller manages virtual instance network and schedules/gathers information about the VM execution on specific node controllers. Storage Controller implements storing and accessing user data and virtual machine images to public storage interface. Cloud Controller is the user and administrator entrant into the cloud. It inquires node managers about resources, brings about high level scheduling decisions, requests to cluster controllers to implement. A Node Controller (NC) is designed to execute on every node that are entitled for VM instances hosting. A NC in response to the controls and queries of Cluster Controller controls the host operating system and the hypervisor on its node. The NC retaliates its queries to identify the physical resources of the nodes (number of cores, size of memory, available disk space) and also to discover the state of the VM instance on the node. The collected information in response to the describe resource and describe instance requests is propagated to the Cluster Controller. The instances may be started as well as stopped through different mechanisms that work way beyond the NCs control. The VM Instances that are in control of the Cluster Controller are restrained to the node's NC making runInstance and terminateInstance request. The NC processes the request with the hypervisor only upon proper authorization verification and confirming the resource availability. For an instance to start, the node-local copy of the instance image files are made either from the remote image repository or from a local cache followed by creating a new endpoint in the virtual network overlay which then instructs the hypervisor to boot up the instance. The Instance images are either from kernel, ram disk or from the root file system. For an instance to stop, the hypervisor is instructed by the NC to terminate the VM, destroys the files related with the instances and tears down the virtual network end point.The Cluster Controller (CC) does the trick only with a cluster front-end machine or machine that probably has network connectivity to both the nodes running NCs and to the machine running the Cloud Controller (CLC). The operations of the Cluster Controller are rather singular than plural which accounts similar to that of NC's operations. (e.g. runInstances, describeInstances, terminateInstances, describeResources). The primitive functions performed by CC encloses scheduling of the incoming instance run requests to specific NCs , by controlling the Virtual Network Overlay instances and also gathers and reports information about sets of NCs. The Cluster Controller uses its describe resource operation to connect with each NC component and sends the run Instances request to the host NC which has enough free resources to host on receiving a set of instances to run. The CC on receiving a describe resources request, it also carries with it a set of resource characteristics (cores, memory, and disk) that describe the requirements of an instance (termed a VM "type") which is used for the calculations on the simultaneous instances of the specific requirement type that can be run on a collection of NCs and this is then reported to the Cloud Controller. System's Resource State (SRS) meant for two stages of execution wherein considering the stage one, the information in the SRS is confided to make an admission control decision when the requests arrive meeting the user-specified service level expectations. The VM creation entails resource reservations in SRS which downstreams the request and commits on success or rollback if errors. The second part is the key source of authority to the properties of running reservations that tracks the state of resource allocations.

Intrusion Prevention Systems:

Intrusion Prevention and Detection systems aims at providing security to computer networks by monitoring processes, identifying and reporting suspicious or malicious activities. A secure environment can be achieved through protection mechanisms such as Intrusion Prevention Systems (IPS) which focuses on the detection of intrusive network. IPS studies and analyse the security of the service delivery models in different aspects of scope, singularity, shows that a greater number of issues pertain in the cloud deployment model. We need to

have an effective security mechanism has to be developed to detect and control a malicious action that intrudes. The elasticity and the distributed architecture of the cloud constrain the IPS to adapt itself to protect the new environment effectively which increases the amount of demanded resources. Therefore, it has to be dynamically expandable. Expansions abide by the consumption demands either by redirection or removal of users. Either it alerts the administrator or triggers counter-measures automatically when it recognizes any malicious actions with the operating system or network. Intrusion Prevention System functions as a security tool such as firewalls and anti-virus software and is primarily designed to enhance the information and communication system's security grades.

Data Stream Management System:

An infinite series of elements that are extracted from an ordered domain constitutes a data stream. The study for a new kind of data called stream-data has been called for by many applications such as sensor network data analysis, network traffic monitoring and web mining, where data sets are stored statically in continuous forms of data, significantly infinite data streams. Difficulties in mining frequent item-sets persist as we are in need for real time responses and also due to the complications in computation as well as mining and updating frequent patterns in a limited window size. For knowledge discovery, data streams are being replaced by Online Mining of frequent items which has become proliferate research challenge. The data elements are first examined over the data streams at least once and the memory usage is restricted while mining (Chang, J., W. Lee, 2004). A quick process of the data elements in the stream is done and the analytical results should be available on the user request. Overall the erroneous results should be reduced to minimal. The algorithms are boundless use only single scan over for knowledge discovery which makes it essential due to the continuous nature of the data streams. Storing all data into main memory formulates it to be unfeasible as it supports both continuous and one-time queries. The correctness of the analytical results can be given upon only by some counting errors in single pass data stream mining algorithms.Basically, knowledge entrenched in a data stream is more likely to be changed as time goes by. Quickly recognizing the recent changes in stream data, especially for online stream data, it is very much useful to provide valuable information for the next level of analysis. Apart from this, monitoring the incessant variation of a data stream enables to find the steady change of in-built knowledge, so that it can be appropriately utilized. This can able achieved by efficiently eliminating the old transactions on the current mining result.

Existing Methods:

Xing *et al*. (2013) proposed a system called SnortFlow for intrusion prevention within cloud environment. The Snortflow prototype is Xen-based cloud built and tested. The snortflow demon component collects the suspicious traffic and generates an alert which is pushed to the interpreter and in turn the rules generator is invoked. The rules for the suspected traffic is generated and then forwarded to the open flow device. Reconfiguration of network is done by the openflow according to the rules that are generated. Thus the Snortflow evaluates and exhibits exceptional performance and intrusion prevention. Zhou *et al*. [6] elaborated extensive survey on privacy and security concerns and a few details on issues related to multi-location storage of various cloud providers were discussed on auditing characteristics and privacy out-dated acts. J. Li, *et al*. (2012) proposed Virtual Network Security is furnished through deploying network devices by Cyberguarder. Utilization of layer-two tunnel VPN between virtual bridges brings about Virtual Isolation. Peer-to-peer transmission of data is carried out. The Central node contains the metadata to optimize the traffic which are monitored by the software ports. Intrusion Detection Systems are adaptive and hence are deployed into the Virtual Networks for the sole purpose of security. The cyberguarder also makes sure of the security issues in the Virtual machines through integrity verification and monitoring system calls. Performance increase of about 10% overhead and Energy Consumption of about 5% increase are the experimental statistics of the system.

Strategy To Prevent Intrusion:

The proposed hybrid method is to combine both data stream mining algorithm techniques and intrusion detection techniques to achieve intrusion prevention. This new method will be used to achieve intrusion prevention and information security can be achieved by an improved standard. To protect the information system from attacking, some detection mechanism are proposed such as based on the analysis of intruder activity and attack approaches, content classification of header, statistical and spectral analysis, and etc. Moreover, some institute might adopt central management with associated distributed defence systems to detect and prevent from intrusion. The goal of anti-intrusion is to protect the information and networking system such as mail servers, database servers, and HTTP servers from paralyzed and ensure the decrease of depletion of bandwidth and decrease the consumption of resource of servers. The experimental architecture is established based on the rationale of centralized management. All the recorded information that collected by system will be analysed by the expert system for integrated analysis. After then, all of the timing based information will be examined by a

Time-sensitive Stream Mined Access to Network as shown in Figure 1.

TIme-sensitive Stream Mined Access to Network (TISMAN) is a technique hidden inside the intrusion prevention architecture which is implicitly represented in the system. As a physical procedure, all of the factors of the recent pre-state that will control the output result in current state will be taken as input streams. The reason that the TISMAN has been adopted for early alert to against intrusion in the paper is because an initiation of intrusion to be performed must trigger a lot of zombies or proxies that have been setting up in advanced. Besides, before initiate a real attacking, some investigation must be performed. Therefore, all of the related features that collected in different timing must be integrated for associated analysis. Instead of allowing all traffic then looking for potential attack signatures, we propose "*Deny All except allowed*" strategy. The oldest transaction in the sliding window will expiry if a new transaction arrives, which is forcing us to create a model to have a new method to find the expired transaction and also to discount the support counts of the item-sets implicated. The fore-most task to be done is to build up a data structure called *discounting table* (DT) to keep hold of frequent item-sets with a support counts for every individual basic block (B) of the present active current *Time sensitive sliding window* (TS). And also a data structure called the *Potentially Frequent-itemset Pool* (PFP) should be maintained to place the frequent item-sets in TS_i and the frequent ones in B_i. Apart from mining and discounting methods, the *self-adjusting discounting table* (SDT) is designed in such a way which can automatically adjust the size of its own while placing the discounting information. With the size constraint of SDT, the item-set will follow an approach to amalgamate the information of more than one item-set placed in SDT. The most important idea behind this design is to reduce the disparity between the unique support count of every item-set and its predictable count after amalgamation. The main finding is that the two guarantees explained above still hold when SDT is deployed. The design consists of the item-sets that are frequent in B_i but not frequent in TS_i-1 in PFP perhaps they are frequent in TS_i.

The design having most important four parameters which should be passed on to the system before its start, the support threshold θ, for every basic block (P) this is the basic unit of time period, the length of TS |W|, and the output mode M. As we mentioned a data stream is been divided in to block which is having divergent number of transaction as per the parameter P. The buffer will be consuming the transaction regularly and dispense them in to our system as block-by-block. This will get discarded directly, once a basic block generates these operations and pass through our system. The threshold array (TA) is used to store the dynamically computed support count threshold for every basic block Bi, because the every basic block will be having distinct number of transaction. The data stream technique which is mounted on network setup as shown in Figure 2.

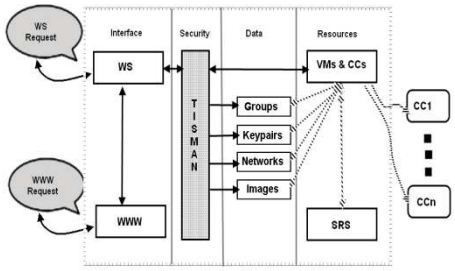

Fig. 1: Illustration of Cloud architecture with TISMAN.

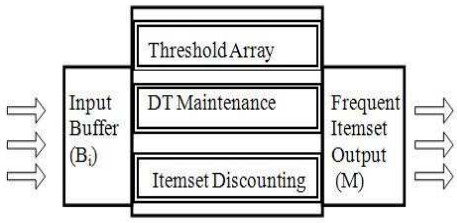

Fig. 2: Data Stream Mining Architecture.

Algorithm 1 – To Find Frequent Itemsets
Input: Stream S, Parameters θ, P, |W|, M
Output: All the frequent itemsets satisfying M
Let TA, PFP, and DT be empty
While B_i comes //*from the buffer*
    If (i = 1)
        New_itemset_insertion
        DT_maintenance
    Else If (i ≤|W|)
        New_itemset_insertion
        Old_itemset_update
        DT_maintenance
    Else
        Itemset_discounting
        New_itemset_insertion
        Old_itemset_update
        DT_maintenance
    TA_update
    Frequent_itemset_output
End While

The above algorithm have been used in this architecture to find the frequent item set stream data and these packet data are send for back tracking or for request denial based on the rule set in the system. The proposed TISMAN algorithm will maintain frequent itemsets over a sliding window. The itemsets generated by TISMAN are maintained in a prefix tree structure, D. An itemset, W, in D has the following three fields: freq(W), err(W) and tid(W), where freq(W) is the frequency of W in the current window since W was inserted into D, err(W) is an

upper bound for the frequency of W in the current window before W was inserted into D, and tid(W) is the ID of the transaction being processed, when W was inserted into D. For each incoming transaction Y with ID = tid_t, TISMAN increments the computed frequency of each subset of Y in D.

Performance Evaluation:

To implement the proposed method in the Eucalyptus cloud-computing framework, we setup a test bed consists of 8 numbers of AMD Quad core computer (2.6 GHz processor, 8 GB RAM) running the Ubuntu Server Linux 12.04.2 LTS and the Eucalyptus System. We have most few customizations in the Ubuntu Linux for operating system for the virtual machines. The customized Eucalyptus architecture framework along with KVM hypervisor is used to build the cloud environment test bed. We started analyzing the utilization of the hardware and software resources in the modified architecture by sending many request from different client and doing all the normal activities and some abnormal activities too, after doing a continuous performance monitoring we can able to find the system still work fine without any disparity.

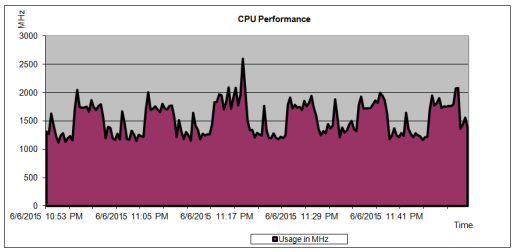

Fig. 3: CPU Performance.

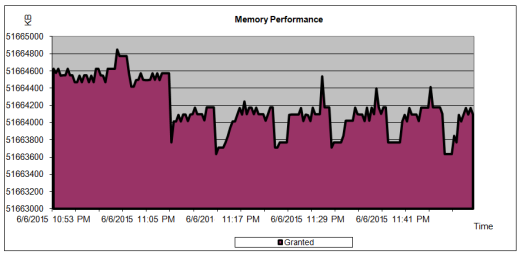

Fig. 4: Memory Performance.

The above graph shows the performance of our server for a time span of an hour within the total testing time period.

Conclusion:

We presented an approach to attain intrusion prevention in Eucalyptus cloud paradigm by modifying its internal architecture which provides us good number of positive results, by combing the time-sensitive sliding window data stream mining model along with cloud architecture, we get new avenue for many research problems. The proposed model is more suitable for high speed processing of massive data streams in real-time. All of the suspected packet information will be composed and moved to the authority module for further analysis. An attacker usually uses one or more programs and each program produces one or more processes. If one process is identified as abnormal, the program containing the process is then classified as abnormal and an intrusion alarm is reported. There are also disadvantages for the models based on the frequency the system will produce the result slowly which may annoy end user.

REFERENCES

Aggarwal, C., Charu, 2007. Data Streams Models and Algorithms: Advances in Database Systems, Vol. 31, Springer, ISBN 978-0-387-28759-1-2.

Chang, J., W. Lee, 2004. A sliding window method for finding recently frequent itemsets over online data streams. Journal of Information Science and Engineering, 20(4): 753–762.

Chih-Hsiang Lin, Ding-Ying Chiu, Yi-Hung Wu, Arbee L.P. Chen, 2005. Mining Frequent Itemsets from Data Streams with a Time-Sensitive Sliding Window. In Proceedings of SDM'.

Subashini, S., V. Kavitha, 2011. A survey on security issues in service delivery models of cloud computing. J. Netw. Comput. Appli. 34(1): 1–11. doi:10.1016/j.jnca.2010.07.006

Xing, T., D. Huang, L. Xu, C. Chung, P. Khatkar, 2013. Snortflow: a openflow-based intrusion prevention system in cloud environment, in: IEEE Research and Educational Experiment Workshop, 89–92.

Zhou, M., R. Zhang, W. Xie, W. Qian, Zhou, 2010. A.: Security and privacy in cloud computing: a survey. In: 6th International Conference on Semantics Knowledge and Grid, 105–112. IEEE Computer Society, Washington, DC, USA.

King, S.T. and P.M. Chen, 2003. “Backtracking Intrusions,” Proc. of the Symposium on Operating Systems Principles (SOSP), 223 – 236, DOI:10.1145/945445.945468

Li, J., B. Li, T. Wo, C. Hu, J. Huai, L. Liu, K.P. Lam, 2012. Cyber-guarder: a virtualization security assurance architecture for green cloud computing, Future Gener. Comput. Syst., 28(2): 379–390.

Kavitha, C., M. Suresh, 2012. Massive stream data processing to attain anomaly Intrusion Prevention, Devices, Circuits and Systems (ICDCS), International Conference on, 3: 572 - 575.

Daniel Nurmi, Rich Wolski, Chris Grzegorczyk Graziano Obertelli, Lamia Youseff, Sunil Soman, Dmitrii Zagorodnov, 2009. The Eucalyptus Open-source Cloud-computing System, Cluster Computing and the Grid, 2009. CCGRID '09. 9th IEEE/ACM International Symposium on, 5: 124-131.