Malicious Behaviour Detection System for Infrastructure as a Service (IaaS) Clouds

S.Jeba Anandh [1], Abinaya.S.P [2], Ganesh Priya.P [3], Keerthana.C [4]
[1] Teaching Faculty, University College of Engineering, Nagercoil, Tamil Nadu, India.
[2, 3, 4] Student, University College of Engineering, Nagercoil, Tamil Nadu, India

***Abstract*— The development of Infrastructure as a Service system brings new opening, which likewise goes with new problems in auto scaling, resource allocation, and security. A fundamental challenge is related to the continuous tracking and monitoring of resource in the framework. In this project, we propose a novel framework to automatically track, monitor, and orchestrate resource usage in an Infrastructure as a Service (IaaS) system that is widely used in cloud. We use a tracking method to continuously track important system usage metrics with low overhead. By using, Principal Component Analysis (PCA) approach to continuously monitor and automatically find anomalies based on the approximated tracking results. We show how to dynamically set the tracking threshold based on the detection results and further, how to adjust tracking algorithm to ensure its optimality under dynamic workloads. We use introspection tools to perform memory forensics on VMs guided by analysed results from tracking and monitoring to identify malicious behaviour inside a VM. The proposed malicious behaviour detection system provides better result for more sophisticated resource orchestration and incorporating the defence against even more complex attacks in previous works.**

Index Terms— Infrastructure as a Service (IaaS), cloud, tracking, monitoring, anomaly detection, virtual machine introspection

1. INTRODUCTION

The Infrastructure as a service framework is a popular model in realizing cloud computing services. A cloud provider manages and outsources her computing resources and cloud users to cut their cost on pay-per-use basis, it has raised new challenges in auto scaling resources and security. Auto scaling is the process to automatically add and remove computing resources based upon the actual resources usage .For example, the Amazon offers cloud service with its Elastic Compute Cloud (EC2) platform. Load balancing and auto scaling is the ability to monitor resource usage from many virtual machines (VMs) running on top of EC2.

Security is another challenges in IaaS framework. For example, it was reported in late July2014, adversaries attacked Amazon cloud by installing distributed denial of services (DDOS) bots on user VMs by exploiting a vulnerability in Elastic search. Resource usage data could provide critical insights to address the security concerns. A cloud provider needs to constantly monitor resource usage and utilize these statistics not only for resource allocation and also detect anomaly in the system. Several applications also need intelligent and automated orchestration of system resources, by going beyond passive tracking and monitoring and introducing auto-detection of abnormal behaviour in the system. Active introspection and correction once anomaly has been identified and confirmed.

2. RELATED WORK

Ke Yiet al. [1] proposed an approach in which competitive online algorithms whose communication costs are compared with the optimal offline algorithm that knows the entire cloud system.These online tracking problems have a variety of application, ranging from sensor monitoring, location-based services, to publish/subscribe systems. This proposed system supports a variety of application, ranging from sensor monitoring, location-based services, to publish/subscribe systems.Data loss and Password leakage occur for security concern due to lack of encryption Heakon Ringberg et al. [2] proposed an approach in which detecting anomalous traffic is a crucial part of managing IP networks, network-wide anomaly detection based on Principal Component Analysis has emerged as a powerful method for detecting a wide variety of anomalies. Achieve such promising early issues because of their great familiarity with both the technique and the data. Anukool Lakhina et al. [3] proposed an approach in which the method is based on a separation of the high-dimensional space occupied by a set of network traffic measurements into disjoint subspaces corresponding to normal and anomalous network conditions. It's diagnose source of the anomaly, but not detect existed anomalies.

Weihua Li et al. [4] proposed an approach in which the algorithms using rank-one modification and Lanczos tridiagonalization, are then proposed and their computational complexity is compared. It's check with only process but not malicious detection. Daniel Moldovan et al. [5] MELA, a customizable framework, which enables service providers and developers to analyse cross-layered, multi-level elasticity of cloud services, from the whole cloud service to service units, based on service structure dependencies. It supports real time multi-level analysis of elastic cloud services. Xin Liet al. [6] proposed an approach which defines how one can use random

aggregations of IP flows (i.e., sketches) to enable more precise identification of the underlying causes of anomalies. This system detect anomalies with high accuracy and identify the IP flows that are responsible for the anomaly.Daniel J et al. [8] proposes PerfCompass, an online performance anomaly fault debugging tool that can quantify whether a production-run performance anomaly has a global impact or local impact.It is light-weight, which makes it practical for use in production cloud infrastructures. Brendan Dolan-Gavitt et al. [4] proposed a technique allows introspection tools to be effortlessly generated for multiple platforms, and enables the development of rich introspection based security applications. The evaluation results clarify that the proposed method achieves requested fault-tolerance level with less number of hosting servers. Yangchuan Fu et al. [6]VMST offers a number of new features and capabilities. Particularly, it automatically enables an in-guest inspection program to become an introspection program. It automatically enables the in-guest inspection program to become an introspection program and largely relieve the procedure of developing customized VMI tools. Benoit Bertholon et al [10] proposed Certicloud -This platform provides the secure storage of users environments and their safe deployment onto a virtualization framework. This mechanism is not supported of Private Cloud Environment. It includes the deeper analysis of CERTICLOUD scalability and its integration into an existing cloud

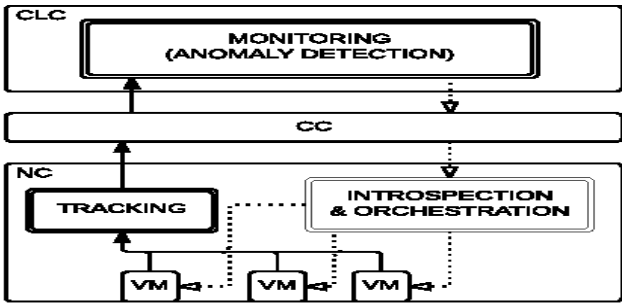
3. PORPOSED MODELLING

The entire proposed modelling and architecture of the current research paper should be presented in this section. This section gives the original contribution of the authors. This section should be written in Times New Roman font with size 10. Accepted manuscripts should be written by following this template. Once the manuscript is accepted authors should transfer the copyright form to the journal editorial office. Authors should write their manuscripts without any mistakes especially spelling and grammar.

4. DESIGNING OBJECTIVE

This monitoring component also adjusts the tracking threshold from the tracking component dynamically online based on the data trends and a desired false alarm rate.Cloud users interact with the cloud controller (CLC) to issue requests such as to allocate resources and query resources usage. CLC handles incoming user requests, collect information of entire cloud, and make high level decision and control. The other components are Cluster Controller (CC)and Node Controller (NC).

A CC forwards requests from the CLC to an NC, gather status data on each NC, and report back to the CLC .An NC controls the VMs running on it. One CLC controls several NCs on which multiple user VMs could be running. Limitation of cloud Watch like AWS is that they only do passive monitoring .No active online resource orchestration is in place towards detecting system anomalies, potential threads and attacks.Active online resource monitoring and orchestration gives the chance to trigger Virtual Machine Introspection (VMI) to debug the system and figure out what has possibly gone wrong.The Introspection into VMs then allows to orchestrate resource usage and allocation in the IaaS system to achieve more secured system and better performance.Our goal is to automate this process and trigger VM Introspection only when needed.



In our system introduces an online tracking module that runs at NC and continuously track various performance matrices and resource usage value of all VMs.The CLC is denoted as tracker and NC is denoted as observer.Our goal is to replace the sampled view at the CLC with a continuous understanding of system status with minimum overhead .The monitoring module is used to continuously monitor resource usage data reported by the online tracking module. The goal is to detect the anomaly by mining the resource usage data.

Virtual Machine Introspection (VMI) is used to detect and identify malicious behaviour inside a virtual machine .IN this techniques such as analysing VM memory space tends to be of great cost. It provide two choice to detect malicious inside the VM.One is set of threshold for each resource usage measure and trigger VMI. In this method already existed in auto scaling process. Another one is online monitoring method in the monitoring module to automatically detect anomaly.

5. MALICIOUS DETECTION

Malware or malicious software, is any program or file that is harmful to a computer user. Malware includes computer viruses,worms,Trojan horses and spyware. These malicious program can perform a variety of function including stealing, encrypting or deleting the sensitive data, altering or hijacking core computing functions or monitoring users computer activity without their permission. In majority of cases, malware is the first point of initiation for large-scale Distributed Denial of Service (DDoS) attacks. Current methods of detecting attacks on cloud infrastructures or the VMs resident within them do not sufficiently address cloud

specific issues. Nevertheless, despite the important lessons learned from these studies they do not develop an overall online detection strategy that considers real-time measurement samples from each VM. Further, these approaches are purely signature based, and as such are not in a position to provide a robust scheme for any future threats posed by novel malware strains due to their simplistic rule-based nature. Each solution to detection is performed in an isolated manner and neglects to consider the unique topology of the cloud, which is at its heart a network of interconnected nodes, each with their own isolated execution environments. If a detection system is to perform effectively within a cloud it is required to possess the capability of communicating detected faults and challenges across the whole infrastructure, especially if it is to perform as part of a larger, autonomous and self-organising, cloud resilience system.

6. ARCHITECTURE AND COMPONENTS

In the architecture (fig:1)of malicious detection, it consist of behaviour analyser of malicious, a system call monitor and a detection engine. The first two components belongs to the training phase and another one is belong to real time detection phase.

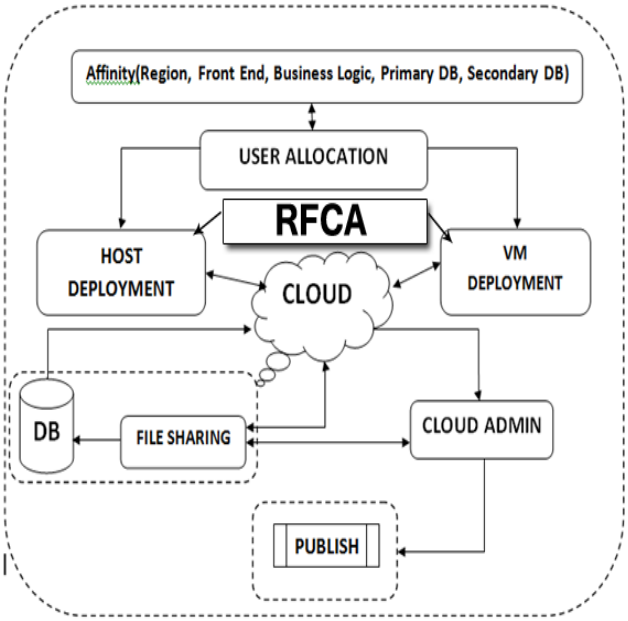


Fig. 1: System Design.

When the user send a request to the server through the internet. The cloud admin first check the ip address and if it's valid reply to the user in the form of needed information to the user. The cloud admin should monitor all the cloud resources by using tracking components. Cloud provider allocate the user in the two types of cloud user provides the access rights to the cloud user.

In virtual machine deployment the user creates user id and password before getting access rights. After that cloud admin generates the session key and it's to the cloud user. Cloud user using that session key creates an account. Then the cloud user creates the host deployment by entering host name, host size in which platform it is used.

The cloud admin creates an account to get the access rights by entering name and password. Then virtual machine and host are mapped. In cloud the database consist of two types of data. The primary database which consist of frequently accessed data's and the secondary database which consist of detailed information. When user needs the access rights to the cloud then the user enters the user id and password and it is verified.

Affinity rules are used to ensure that the desired virtual server or workload is hosted on the target host. When measured activity is outside baseline parameters or clipping level. A profile for a network might show that web activity compromises an averages of 13% of network bandwidth at the internet border during typical workday hours

7. COMPONENTS

A. Placement Constraint Extraction:

Virtual machine placement is the process of selecting the most suitable host for the virtual machine. The process involves categorising the virtual machines hardware and resources requirements and the anticipated usage of resources and the placement goal. The placement goal can either be maximizing the usage of available resources or it can be saving of power by being able to shut down some servers. The autonomic virtual machine placement algorithms are designed keeping in mind the above goals. Placement constraints between different VM types and those between VM types and specific named locations can be extracted from the service structure graph.

B.Tracking Component:

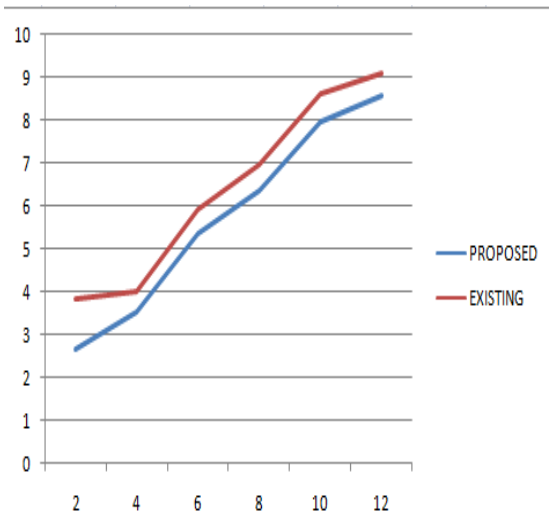
A user VM in Eucalyptus is called an instance. In the following we will use the term "instance" and "VM" interchangeably. There are various variables that can be monitored overtime on each instance, each of which is called a metric. The measurement for each metric, for example, Percent for CPU Utilization, Count for DiskReadOps and DiskWriteOps, Bytes for DiskReadBytes, DiskWriteBytes, NetworkIn and NetworkOut, is called Unit and is numerical. A continuous understanding of these values is much more useful than a periodic, discrete sampled view that are only available, say, every minute. But doing so is expensive; an NC needs to constantly sending data to the CLC. A key observation is that, for most purposes, cloud users may not be interested in the exact value at every time instance. Thus, a continuous understanding of these values within some

predefined error range is an appealing alternative. For example, it's acceptable to learn that CPU Utilization is guaranteed to be within ±3% of its exact value at any time instance.

C .Monitory Component:

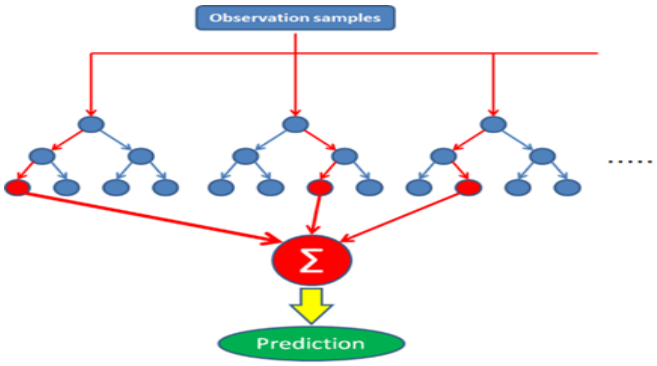
In communication complexity, Alice has x and Bob has y, and the goal is to compute some function f(x,y) by communicating the minimum number of bits between them. There are two major differences between communication complexity and online tracking: First, in online tracking, only Alice sees the input, Bob just wants to keep track of it. Secondly, in communication complexity both inputs x and y are given in advance, and the goal is to study the worst-case communication between x and y; while in online tracking, the inputs arrive in an online fashion, and we focus on the competitive ratio.

It is easy to see that the worst-case (total) communication bounds for our problems are mostly trivial. In data streams [1], the inputs arrive online, and the goal is to track some function over the inputs received so far. In this aspect it is similar to our problem. However, the focus in streaming algorithms is to minimize the space used by the algorithm, not communication. The memory contents could change rapidly, so simply sending out the memory contents could lead to high communication costs.



c.Random Forest Machine Learning Algorithm to detect Malicious:

This algorithm detect abnormal behaviour system. Random Forest algorithm to train data for malicious system. This algorithm combination of decision trees that depend on the sampled random vector. The collected features are represented as a vector with the train data that has been obtained from collected data.



Algorithm for random forest machine learning

Precondition: A training set S := (x1, y1), . . . ,(xn, yn), features F, and number of vm in host

function

RandomForest(S , F)

1. H ← ∅
2. for i ∈ 1, . . . , B do
3. S (i) ← A bootstrap sample from S
4. hi ← RandomizedVMLearn(S (i) , F)
5. H ← H ∪ {hi}
6. end for
7. return H
8. end function
9. function RandomizedVMLearn(S , F)
10. At each node:
11. f ← very small subset of F
12. Split on best feature in f
13. return The learned tree
14. end function

8. EVALUATION

In the evaluation online tracking function that have a communication complexity. Here we consider observer has x and tracker has y to compute some function that is f(x,y) by communicating the minimum number of bits between them.There are two major difference between communication complexity and online tracking.

1. The observer has x sees the input, y just wants to keep track of it.

2. Both inputs x and y are given in advance.We use in online tracking .The inputs are arrived in an online fashion and focus competitive ratio. The inputs are arrived in an online in the

form of data stream. This streaming algorithm is to minimize the space used by algorithm not communication. The memory contents could change rapidly and simply sending out the site memory contents could lead to high communication costs. In distributed tracking, the inputs are distributed among multiple sites and arrive online. Both online and offline algorithm can only communicate .When the error can be allocated to some site is violated and the can only send in the current value t

9. CONCLUSION

In this project we implemented the security issues in cloud infrastructure as a service, our main focus is on monitoring and detecting abnormal behaviour in cloud infrastructure and orchestration of infrastructure. We present the framework that can be easily integrated into a standard IaaS system to provide automated, continuous tracking, monitoring, and orchestration of system resource usage in nearly real-time. The empirical results that the cloud infrastructure built on the given architecture is affecting in providing security to the communications in the cloud infrastructure.

REFERENCES

[1] Ke Yi, Qin Zhang "Multidimensional Online Tracking " ACM Transactions on Algorithms(Volume 8 Issue 2, April 2012 Article No. 12)

[2] Heakon Ringberg, Augustin Soule, Jennifer Rexford, Christophe Diot "Sensitivity of PCA for traffic anomaly detection," ACM SIGMETRICS international conference on Measurement and modeling of computer systems(Volume 35 Issue 1, June 2007)

[3] Anukool Lakhina, Mark Crovella, Chritophe Diot "Diagnosing network-wide traffic anomalies," SIGCOMM '04(Volume 34 Issue 4, October 2010)

[4] Weihua Li, H.Henry Yue, Sergio Valle-Cervantes, S.Joe Qin "Recursive PCA for adaptive process monitoring," Journal of Process Control 2010 – ELSEVIER.

[5] Daniel Moldovan, Georgiana Copil, Hong-Linh Truong, Schahram Dustdar "MELA: Monitoring and Analyzing Elasticity of Cloud Servic," IEEE International Conference on Cloud Computing Technology and Science(Vol. 2, No. 1, 2015)

[6] Xin Li, Fang Bian, Mark Crovella, Christophe Diot "Detection and Identification of Network Anomalies Using Sketch Subspaces," SIGCOMM conference on Internet measurement (October 25 - 27, 2006)

[7] Daniel J, Hiep Nguyen, Peipei Wang, Xiaohui Gu, Anca Sailer, Andrzej Kochut "PerfCompass: Online Performance Anomaly Fault Localization and Inference in Infrastructure-as-a-Service Clouds," IEEE Transactions on Parallel and Distributed Systems(Issue No.01,vol.2015)

[8] Brendan Dolan-Gavitt, Tim Leek, Michael Zhivich, Jonathon Giffin, Wenke Lee "Virtuoso: Narrowing the Semantic Gap in Virtual Machine Introspection," 2011 IEEE Symposium on Security and Privacy.

[9] Yangchun Fu, Zhiqiang Lin "Space Traveling across VM: Automatically Bridging the Semantic Gap in Virtual Machine Introspection via Online Kernel Data Redirection," IEEE Symposium On Security and Privacy

[10] Benoit Bertholon, Sebastien Varrette, Pascal Bouvry "CERTICLOUD: a Novel TPM-based Approach to Ensure Cloud IaaS Security," 2011 IEEE 4th International Conference on Cloud Computing.