Determination and Prevention of Pernicious activity in Cloud Computing Environment

Shaishav R. Shah
LJIET, GTU
Ahmedabad, India
Shah.shaishav@outllook.com

***Abstract*—Cloud computing environment is a pool of different services, which deliver services in an economical way. However, for many security sensitive applications such as critical data processing, we must provide necessary security protection for migrating those critical application services into shared open cloud infrastructures. In this paper I present a framework to assure the pernicious data processing in cloud Computing Environment. This framework provide stronger attacker pinpointing power. In this pernicious service provider based on dynamically derived trust scores.**

Keywords—Cloud; Security; Secure Dataflow Processing; Serice Integrity

I. INTRODUCTION

Cloud System [1] have recently emerged as popular resource Leasing Infrastructure. Application service providers can lease a set of resources from the cloud system, without investing for their own infrastructures. Cloud systems are particularly amenable for data processing services [2], which are often extremely resource-intensive.

Data-intensive computing has recently received much research attention with many real world application such as security surveillance, specific study and business intelligence in particular, our work focuses on dataflow processes systems that provide high performance continues Processing over massive data streams. Attacks can also pretend to be legitimate service providers to compromise dataflow processing. One of the top security concerns for cloud users to verify the integrity of data processing results, especially for critical data processing applications such as fraud detection and business intelligence.

In this paper, I present the new integrated framework for multitenant cloud system; my framework builds upon previous work IntTest[8], RunTest[7] and AdapTest[6]. First come RunTest it is lightweight application level scheme that can dynamically verify the integrity of Data processing results in the cloud infrastructure and pinpointing a malicious serivce providers when inconsistent result detected. Then comes AdapTest[6] is novel adaptive runtime service integrity attestation framework for large-scale cloud systems. AdapTest builds on top of our previously developed system RunTest[7] that performs randomized probabilistic attestation and employs a clique-based algorithm to pinpoint malicious nodes. AdapTest[6] dynamically evaluates the trustiness of different services based on previous attestation results and adaptively selects attested services during attestation. AdapTest dynamically derives a set of trust scores to achieve differentiated probabilistic attestation. Than third modified scheme is IntTest takes a holistic approach by systematically examining both consistency and inconsistency relationships among different service providers within the entire cloud system[8]. IntTest examines both per-function consistency graphs and the global inconsistency graph. The per-function consistency graph analysis can limit the scope of damage caused by colluding attackers, while the global inconsistency graph analysis can effectively expose those attackers that try to compromise many service functions[8].

In my work I validate aggregation analysis, which provide tightly coupled group of service providers, which contains same service functions in form of Binary search tree. In which we compare the output of the result with adjacent service functions (compare result with one Parent and two child) by executing same task on those nodes of the tree. If there is any inconsistency among them then trust of the node will be decrease as per result and then it have to lost its position in tree. And it has to set on lower level of the tree. Like Parent become a child and child become a parent.

By taking Binary search Tree structure approach I can not only pinpoint attackers more efficiently, also provides result auto correction that can automatically replace corrupted data processing result produced by pernicious service provider with benign service Providers. And Trustiness of the service provider is stored in CSV file, which located at the server. Also mapping of the tree stored in another CSV file. And than it will used to create tree structure. Basically tree structure is used to monitoring of trustiness of service provider on specific services service function and for comparing the result of the service provider.

Specifically, this paper makes the following contributions:

- I provide a scalable and efficient distributed service integrity framework for large-scale cloud computing environment.
- I present a Tree based service integrity detection scheme that can achieve higher pinpointing accuracy than previous techniques.

- We describe a result auto correction technique that can automatically correct the corrupted results produced by pernicious attackers.
- I use file system instead of database for storing temporary data of trustiness and fault detection rate.

I have implemented this scheme on cloudsim. The rest of paper organized as follows: Section 2 presents my system model. Section 3 presents design details. Section 4 provides an study about our scheme. Section 5 presents the experimental results. Finally paper concludes in section 6.

II. PRELIMINARY

In this section, we first introduce the software-as-a-service cloud system model. We then describe our problem formulation including the service integrity attack model and our key assumptions.

A. SaaS Cloud System Model

SaaS cloud builds upon the concepts of software as a service [3] and service-oriented architecture [4], [5], which allows application service providers to deliver their applications via large-scale cloud computing infrastructures. For example, both Amazon Web Service and Google AppEngine provide a set of application services supporting enterprise applications and big data processing.

In a large-scale SaaS cloud, different ASPs can provide the same service function. Those functionally equivalent service components exist because: 1) service providers may create replicated service components for load balancing and fault tolerance purposes; and 2) popular services may attract different service providers for profit. Which service functions are provided by which service providers in the SaaS cloud. Neither cloud users nor individual ASPs have the global knowledge about the SaaS cloud such as the number of ASPs and the identifiers of the ASPs offering a specific service function.

III. PROBLEM FORMULATION

Given a SaaS cloud system; the goal is to pinpoint any malicious service provider that offers an untruthful service function. Which does not require any special hardware or secure kernel support on the cloud platform. We now describe our attack model and our key assumptions as follows:

A. Attack Model

A malicious attacker can pretend to be a legitimate service provider or take control of vulnerable service providers to provide untruthful service functions. The stealthy behavior makes detection more challenging due to the following reasons: 1) the detection scheme needs to be hidden from the attackers to prevent attackers from gaining knowledge on the set of data processing results that will be verified and therefore easily escaping detection; and 2) the detection scheme needs to be scalable while being able to capture misbehavior that may be both unpredictable and occasional. In a large-scale cloud system, we need to consider colluding attack scenarios where multiple malicious attackers collude or multiple service sites are simultaneously compromised and controlled by a single malicious attacker. Attackers could sporadically collude, which means an attacker can collude with an arbitrary subset of its colluders at any time. Attackers can also change their attacking and colluding strategies arbitrarily.

B. Assumptions

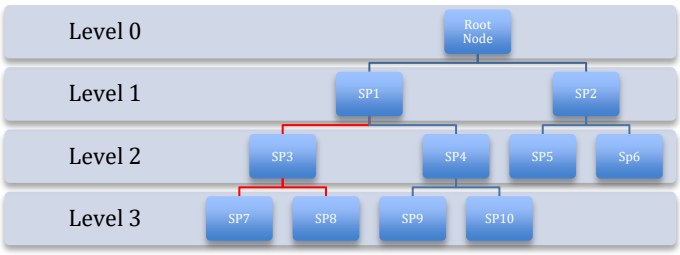
- Cloud have own trusted Service Functions, which provide by all the third party service providers.

IV. DESIGN AND METHODOLOGY

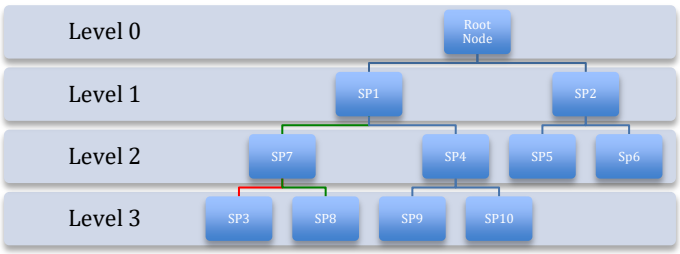
In this section, I present the basis of my approach or framework: probabilistic Tree Data structure for decide which two service functions will check their results. Then on basis of their consistency/inconsistency we can decide the trust of the service function of service of service provider.

In my work I will create a tree based on their trust. Then while attempting a service by user at that time its output compare with their child node. When it is compare with it's child node if unlatch in result found then result of child node of object node is compared with the result of parent node of object node. if result match with each other then trust of object node is decrease by one . and it loose its position in the tree and it have to traverse on the lover level of the tree.

Before correction



After Correction



A. Result Autocorrection

1) My work can not only pinpoint malicious service providers but also automatically correct corrupted data processing results to improve the result quality of the cloud data processing service.

2) When the output / result mismatch each other then it will be the replace after comparing third high trusted node or Parent node.

V. SECURITY ANALYSIS

Although our scheme guarantee zero false positives even though there are multiple independent colluding groups, it will be difficult for attackers to escape our detection with multiple independent colluding groups since attackers will have inconsistency links not only with benign nodes but also with other groups of pernicious nodes. Additionally, our approach limits the damage colluding attackers can cause if they can evade detection in two ways. First, our scheme limits the number of functions, which can be simultaneously attacked. Second, our approach ensures a single attacker cannot participate in compromising an unlimited number of service functions without being detected.

VI. CONCLUSION

In this paper I have presented framework to verify the integrity of distributed system. Furthermore, it provides result autocorrection to automatically correct compromised results to improve the result quality.

REFERENCES

[1] Amazon Elastic Compute Cloud. http://aws.amazon.com/ec2/.

[2] Juan Du, Xiaohui Gu "Adaptive Data-Driven Service Integrity Attestation for Multi-Tenant Cloud System" Bill Lin University of California, San Diego IEEE Press Piscataway, NJ, USA ©2011J.

[3] Zhou Li, Sumayah Alrwais,Yinglian Xie, Fang YuXiaoFeng Wang On this and that. Finding the Linchpins of the Dark Web: a Study on Topologically Dedicated Hosts on Malicious Web Infrastructures ISSN : 1081-6011

[4] G. Geethakumari, Abha Belorkar On this and that. Regenerating Cloud Attack Scenarios using LVM2 based System Snapshots for Forensic Analysis ISSN: 2089-3337

[5] Hung-Jen Liao , Chun-Hung Richard Lin , Ying-Chih Lin , Kuang-Yuan Tung On this and that. Intrusion detection system: A comprehensive review ISSN: 1084-8045

[6] Juan Du, Wei Wei, Xiaohui Gu, and Ting Yu On this and that. RunTest: Assuring Integrity of Dataflow Processing innCloud Computing Infrastructures ISSN: 978-1-60558-936-7

[7] Juan Du, Xiaohui Gu On this and that. Adaptive Data-Driven Service Integrity Attestation for Multi-Tenant Cloud Systems ISSN: 978-1-4577-0103-0

[8] Juan Du, Daniel J. Dean, Yongmin Tan, Xiaohui Gu, Ting Yu On this and that. Adaptive Data-Driven Service Integrity Attestation for Multi-Tenant Cloud Systems ISSN: 1045-9219