Open access Journal International Journal of Emerging Trends in Science and Technology

An Attestation of Service Integrity for SaaS Clouds using IntTest method

Authors
Praveen Kumar P S[1], Prasannakumar.M[2]
[1]PG Scholar, Sridevi Institute of Engineering and Technology, Tumkur
Email: *Pspraveen013@gmail.com*
[2]Asst.Professor, Sridevi Institute of Engineering and Technology, Tumkur
Email: *Prasan.ctn19@gmail.com*

Abstract
Software-as-a-service (SaaS) cloud systems make application service providers to deliver their applications via massive cloud computing infrastructures. However, due to their sharing nature, SaaS clouds are vulnerable to malicious attacks. In this paper, we present IntTest, a scalable and effective service integrity attestation framework for SaaS clouds. IntTest provides a novel integrated attestation graph analysis scheme that can provide stronger attacker pinpointing power than previous schemes. Moreover, IntTest can automatically enhance result quality by replacing bad results produced by malicious attackers with good results produced by benign service providers. Our experimental results show that IntTest can achieve higher attacker pinpointing accuracy than existing approaches. IntTest does not require any special hardware or secure kernel support and imposes little performance impact to the application, which makes it practical for large-scale cloud systems.

1. Introduction

CLOUD computing has developed as a cost-effective resource leasing paradigm, which obviates the need for users maintain complex physical computing infrastructures by themselves. Software-as-a-service (SaaS) clouds (e.g., Amazon Web Service (AWS) and Google App Engine) build upon the concepts of software as a service and service-oriented architecture (SOA) which enable application service providers (ASPs) to deliver their applications via the massive cloud computing infrastructure. In particular, our work focuses on data stream processing services that are considered tobe one class of killer applications for clouds with many real-world applications in security surveillance, scientific computing, and business intelligence. However, cloud computing infrastructures are often shared by ASPs from different security domains, which make them vulnerable to malicious attacks.

Moreover, service integrity is the most prevalent problem, which needs to be addressed no matter whether public or private data are processed by the cloud system. Although previous work has provided various software integrity attestation solutions those techniques often require special trusted hard ware or secure kernel support, which makes them difficult to be deployed on large-scale cloud computing infrastructures.

In this paper, we present IntTest, a new integrated service integrity attestation framework for multitenant cloud systems. IntTest provides a practical service integrity attestation scheme that does not assume trusted entities on third-party service provisioning sites or require application modifications. IntTest builds upon our previous work RunTest and AdapTest but can provide stronger malicious attacker pinpointing power than RunTest and AdapTest. Specifically, both Run Text and Adap Test as well as traditional majority voting schemes need to assume that benign service providers take majority in every service function. However, in large-scale

multitenant cloud systems, multiple malicious attackers may launch colluding attacks on certain targeted service functions to invalidate the assumption. To address the challenge, IntTest takes a holistic approach by systematically examining both consistency and inconsistency relationships among different service providers within the entire cloud system. 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. Hence, IntTest can still pinpoint malicious attackers even if they become majority for some service functions.

2. Related work

In recent years many integrity attestation schemes have been developed for software as a service clouds. For example the BIND technique, AdapTest technique, RunTest technique etc. but all of these are having some problems some of them needs secure kernel support and special trusted hardware components. In BIND (Binding Information and Data) technique is a verification method of integrity services that are provided by the software as a service cloud system. It was a fine grained attestation framework and can provide the verification through a secure kernel or by a third party. This technique uses the following steps: 1) attestation annotation mechanism 2) sandbox mechanism 3) verification of authenticator through hash. BIND method uses the Diffee- Hellman key exchange for the purpose of integrity attestation. Another existing technique is TEAS (Timed Executable Agent System) this is used for protecting the integrity of cloud computing platforms. An agent generation and verification algorithm is used in this TEAS method.

Another one existing technique is the RunTest, it is a scalable runtime integrity attestation framework. It provides a light weight application level attestation method to assure the integrity of data flow processing in cloud. This will identify the untruthful data flow processing and will pinpoint malicious data processing service provider and at last it will detect the attackers behaviour. This RunTest will provide the benign service providers and will determine the malicious behavior of the attackers. But the disadvantage is its low performance.

The AdapTest is another one existing technique, it provides a novel adaptive data driven runtime service integrity attestation framework. This method will significantly reduce the overhead of attestation and will shorten the delay. It treats all components as black boxes and it does not need any special hardware or software requirements. In this AdapTest it will reduce the attestation overhead and the detection of malicious attackers or service providers will be high when compared to other techniques. All the above methods that are used in the existing papers are having some disadvantages and to overcome that disadvantages this IntTest is using. And by using this IntTest it will provides more integrity and it will provide more accuracy in pinpointing the malicious attackers and service providers. Also it will provide a result auto correction method and will correct the bad results and replace it with good results and also in this it doesnot require any special hardware and secure kernel support.

3. Proposed System

Software as a service and service oriented architecture are the basic concepts of SaaSclouds and this will allow the application service provider to deliver their application via cloud computing infrastructure. In our proposed method we are introducing a new concept called IntTest. The main objective of IntTest is, it can pinpoint all the malicious service providers. IntTest will treat all the service providers as black boxes and this does not need any special hardware or secure kernel

support. When we are considering the large scale cloud system multiple service providers may simultaneously compromised by a single malicious attacker. In this we assume that the malicious nodes are not having any knowledge about the other nodes except those which they are directly interacting.

In this proposed system we are making some assumptions. First of all we are assuming that the total number malicious service components are less than that of the total number of benign service providers in the entire cloud. This assumptions is very important because without this assumption, it would be difficult for any attack detecting scheme to work successfully. The second assumption is the data processing services are important deterministic. That is, the same input that are giving by a benign service component will always produce the same output and finally we assume that the inconsistency caused by hardware or software faults can be excluded from malicious attacks.

In this section we present the main modules in the proposed system. It mainly consists of four modules that are described below.

A. Baseline Attestation Scheme

IntTest is used to detect the service integrity attack and to pinpoint malicious service providers. For that first we are deriving the consistency and inconsistency relationship between service providers. Consider the fig.2 it shows the consistency check method. In that p1,p2 and p3 are the service providers. All of them offers the same function f. The portal sends the original data d1 to the service providers p1 and gets the processing result f (d1). Then the portal sends the duplicate of d1 to p3 and gets the result f(d1'). And if both of them are same means it is consistent and if not means they are inconsistent that is if two service providers disagree with each other, when processing the same input then any one of them will be malicious. Thus the malicious attackers cannot escape from detecting when they are providing bad results with good results.

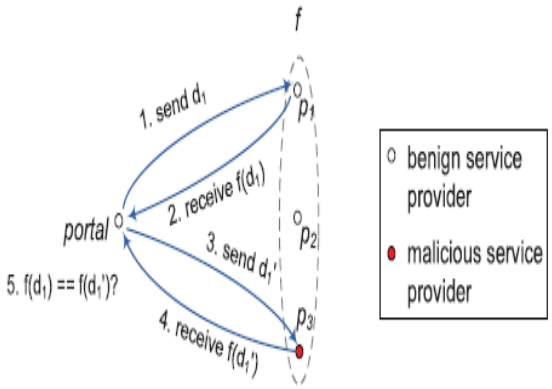

Fig: Consistency check

B. Integrated Attestation Scheme

Here we present an integrated attestation graph analysis algorithm.

Step 1: Consistency analysis: In the first step it will examine the per-function consistency graph and will pinpoint suspicious service providers. The consistency links in the consistency graph will provide a set of service providers. It will keep consistent with each other on a specific service function. The benign service providers will always keep consistent with each other and will form a clique in terms of consistency links. The colluding attackers can try to escape from being detected. Then next we must examine the per-function in consistency graph too.

Step 2: Inconsistency analysis: This inconsistency graph will contain only the inconsistency links, this may exist in different possible combinations of the benign node and the malicious node set. First we assume that the total number of malicious service providers in the cloud system is not more than the benign service providers, then we can pinpoint a set of malicious service providers. If two service providers are connected by an inconsistency link, we can say that any one of them is malicious.

4. Result auto correction

IntTest 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, illustrated by Fig. 3. Without our attestation

scheme, once an original data item is manipulated by any malicious node, the processing result of this data item can be corrupted, which will result in degraded result quality. IntTest leverages the attestation data and the malicious node pinpointing results to detect and correct compromised data processing results.

Specifically, after the portal node receives the result f(d) of the original data d, the portal node checks whether the data d has been processed by any malicious node that has been pinpointed by our algorithm. We label the result f(d) as "suspicious result" if d has been processed by any pinpointed malicious node. Next, the portal node checks whether d has been chosen for attestation. If d is selected for attestation, we check whether the attestation copy of d only traverses good nodes. If it is true, we will use the result of the attestation data to replace f(d). For example, in Fig. 3 the original data d are processed by the pinpointed malicious node s6, while one of its attestation data d'' is only processed by benign nodes. The portal node will use the attestation data result f(d'')to replace the original result that can be corrupted if s6 cheated on d.

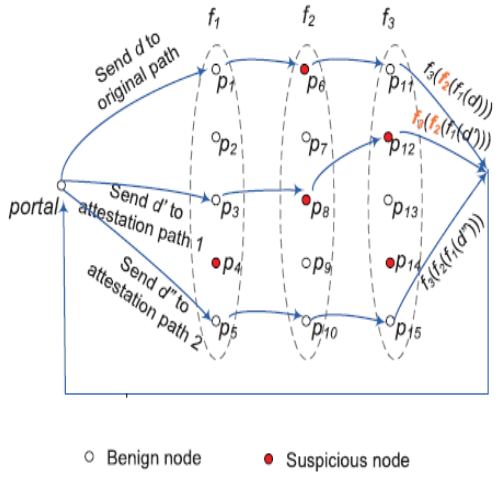

Fig: Automatic data correction using attestation data processing results

This technique can achieve higher detection rate than any other existing technique and will have low false alarm rate than others. Also IntTest can achieve higher detection accuracy than any other techniques when malicious service providers attack more nodes. This method will identify the attackers even though they attack a very low percentage of services.

5. Conclusion

In this paper we introduced a novel integrated service integrity attestation graph analysis scheme for multitenant software-as-a-service cloud system. IntTest uses a reply based consistency check to verify the service providers. IntTest will analyses both the consistency and inconsistency graphs to find the malicious attackers efficiently than any other existing techniques. And also it will provide a result auto correction to improve the result quality.

References

1. Juan Du, Daniel J. Dean, Yongmin Tan, XiaohuiGu, and Ting Yu "Scalable Distributed Service Integrity Attestation for Software-as-a-Service Clouds" IEEE Transactions On Parallel And Distributed Systems, Vol. 25, No. 3, March 2014
2. Du.J, Wei.W, Gu.X, and Yu.T, "Runtest: Assuring Integrity of Dataflow Processing in Cloud Computing Infrastructures," Proc.ACMSymp. Information, Computer and Comm. Security (ASIACCS),2010.
3. Du.J, Shah.N, and Gu.X, "Adaptive Data-Driven Service Integrity Attestation for Multi-Tenant Cloud Systems," Proc. Int'l Workshop Quality of Service (IWQoS), 2011. Virtual Computing Lab, http://vcl.ncsu.edu/, 2013.
4. Shi.E, Perrig.A, and Doorn.L.V, "Bind: A fine-grained attestation service for secure distributed systems," in Proceedings of the IEEE Symposium on Security and Privacy, 2005.