ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Scalable Distributed Service Integrity Attestation for Software-As-A-Service Clouds

K.Bhavani Sankar
P.G. Scholar (M. Tech),
Department of CSE,
Srinivasa Institute of Technology & Sciences,
Ukkayapalli, Kadapa, Andhra Pradesh.

K.Rajasekhar Reddy
Assistant Professor,
Department of CSE,
Srinivasa Institute of Technology & Sciences,
Ukkayapalli, Kadapa, Andhra Pradesh.

ABSTRACT:

Software-as-a-service (SaaS) cloud systems enable 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. We have implemented a prototype of the IntTest system and tested it on a production cloud computing infrastructure using IBM System S stream processing applications. 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.

Index Terms:

Distributed service integrity attestation, cloud computing, secure distributed data processing.

1. INTRODUCTION :

CLOUD computing has emerged 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) [1] and Google AppEngine [2]) build upon the concepts of software as a service [3] and service-oriented architecture (SOA) [4], [5], 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 [6], [7], [8] that are considered to be 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. For example, attackers can pretend to be legitimate service providers to provide fake service components, and the service components provided by benign service providers may include security holes that can be exploited by attackers. Our work focuses on service integrity attacks that cause the user to receive untruthful data processing results. Although confidentiality and privacy protection problems have been extensively studied by previous research service integrity attestation problem has not been properly addressed. 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 hardware or secure kernel support, which makes them difficult to be deployed on large-scale cloud computing infrastructures. Traditional Byzantine fault tolerance (BFT) techniques can detect arbitrary misbehaviours using full-time majority voting (FTMV) over all replicas, which however incur high overhead to the cloud system. A detailed discussion of the related work can be found in Section 5 of the online supplementary material, which can be found on the Computer SocietyDigital Libraryathttp://doi.ieeecomputersociety.org/10.1109 TPDS.2013.62. IntTest, a new integrated service integrity attestation framework for mult itenant cloud systems. IntTest p rovides a practical service integrity attestation scheme that does not assume trusted entities on third-party service provisioning sites or require applicat ion modifications. IntTest builds upon our previous work RunTest and AdapTest but can provide stronger malicious attacker pinpointing power than RunTest and AdapTest.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Specifically, both RunText and Adap Test as well as tradit ional majority voting schemes need to assume that benign service providers take majority in every service function. However, in large-scale multitenant cloud systems, mult iple 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 daa flow processing in cloud. This will will identify the untruthful data flow processing and will pinpoint mallicious data processing service provider and atlast it will detect the attackers behaviour. This RunTest will provide the benign service providers and will determine the malicious behaviour of the attackers.

But the disadvantage is its low performance. The AdapTet is another one existing technique, it provides a novel adaptive data driven runtime service integrity attestation frmaework. This method will significantly reduce the overhead of attestation and will shoten 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 hardwares and secure kernel support..

EXISTING SYSTEM:

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 to be 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. For example, attackers can pretend to be legitimate service providers to provide fake service components, and the service components provided by benign service providers may include security holes that can be exploited by attackers.

DISADVANTAGES OF EXISTING SYSTEM:

•Those techniques often require special trusted hardware or secure kernel support.

•Which makes them difficult to be deployed on large-scale cloud computing infrastructures.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

3 . PROPOSED SYSTEM:

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 RunText and AdapTest 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.

4. SAAS CLOUD SYSTEM MODEL :

SaaS cloud builds upon the concepts of software as a service [3] and service-oriented architecture .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. A distributed application service can be dynamically composed from individual service components provided by different ASPs (pi). For example, a disaster assistance claim processing application consists of voice-over-IP (VoIP) analysis component, e-mail analysis component, community discover component, and clustering and join components. Our work focuses on data processing services .which have become increasingly popular with applications in many real-world usage domains such as business intelligence, security surveillance, and www.jreecs.com scientific computing. Each service component, denoted by ci, provides a specific data processing function, denoted by fi, such as sorting, filtering, correlation, or data mining utilities. Each service component can have one or more input ports for receiving input data tuples, denoted by di, and one or more output ports to emit output tuples . In a large-scale SaaS cloud, the same service function can be provided by different ASPs.

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. To support automatic service composition, we can deploy a set of portal nodes that serve as the gateway for the user to access the composed services in the SaaS cloud. The portal node can aggregate different service components into composite services based on the user's requirements. For Fig. 1. Service integrity attack in cloud-based data processing. Si denotes different service component and VM denotes virtual machines. security protection, the portal node can perform authentication on users to prevent malicious users from disturbing normal service provisioning. Different from other open distributed systems such as peer-to-peer networks and volunteer computing environments, SaaS cloud systems possess a set of unique features. First, third-party ASPs typically do not want to reveal the internal implementation details of their software services for intellectual property protection. Thus, it is difficult to only rely on challenge-based attestation schemes where the verifier is assumed to have certain knowledge about the software implementation or have access to the software source code. Second, both the cloud infrastructure provider and third-party service providers are autonomous entities. It is impractical to impose any special hardware or secure kernel support on individual service provisioning sites. Third, for privacy protection, only portal nodes have global informat ion about 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.

ADVANTAGES OF PROPOSED SYSTEM:

•A scalable and efficient distributed service integrity attestation framework for largescale cloud computing infrastructures.

•A novel integrated service integrity attestation scheme that can achieve higher pinpointing accuracy than previous techniques.

•A result autocorrection technique that can automatically correct the corrupted results produced by malicious attackers.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

•Both analytical study and experimental evaluation to quantify the accuracy and overhead of the integrated service integrity attestation scheme.

4 .DESIGN AND ALGORITHMS :

In this section, first present the basis of the IntTest system: probabilistic replay-based consistency check and the integrity attestation graph model and then describe the integrated service integrity attestation scheme in detail. Next, present the result auto correction scheme.

4.1 Baseline Attestation Scheme :

To detect service integrity attack and pinpoint malicious service providers, our algorithm relies on replay-base consistency check to derive the consistency/inconsistenc relationships between service providers. For example, Fig. shows the consistency check scheme for attesting thre service providers p1, p2, and p3 that offer the same service function f. The portal sends the original input data d1 to p and gets back the result fðd1Þ. Next, the portal sends d01, duplicate of d1 to p3 and gets back the result fðd01Þ. Th portal then compares fðd1Þ and fðd01Þ to see whether p1 and p3 are consistent The intuition behind our approach is that if two service providers disagree with each other on the processing result of the same input, at least one of them should be malicious. Note that we do not send an input data item and its duplicates (i.e., attestation data) concurrently. Instead, we replay the attestation data on different service providers after receiving the processing result of the original data.Thus, the malicious attackers cannot avoid the risk of being detected when they produce false results on the original data. Although the replay scheme may cause delay in a Fig. 2. Replay-based consistency check. single tuple processing, we can overlap the attestation and normal processing of consecutive tuples in the data stream to hide the attestation delay from the user. If two service providers always give consistent output results on all input data, there exists consistency relationship between them. Otherwise, if they give different outputs on at least one input data, there is inconsistency relationship between them. We do not limit the consistency relationship to equality function since two benign service providers may produce similar but not exact ly the same results. For example, the credit scores for the same person may vary by a small difference when obtained from different credit bureaus. Also allow the user to define adistance function to quantify the biggest tolerable result difference.

4.2 Integrated Attestation Scheme:

Here now present our integrated attestation graph analysis algorithm. Step 1: Consistency graph analysis. We first examine perfunction consistency graphs to pinpoint suspicious service providers. The consistency links in perfunction consistency graphs can tell which set of service providers keep consistent with each other on a specific service function. Given any service function, since benign service providers always keep consistent with each other, benign service providers will form a clique in terms of consistency links. For example, in Fig. 3a, p1, p3 and p4 are benign service providers and they always form a consistency clique. In our previous work , we have developed a clique-based algorithm to pinpoint malicious service providers. If we assume that the number of benign service providers is larger than that of the malicious ones, a benign node will always stay in a clique formed by all benign nodes, which has size larger than bk=2c, where k is the number of service providers provisioning the service function. Thus, we can pinpoint suspicious nodes by identifying nodes that are outside of all cliques of size larger than bk=2c are excluded from the clique of size 3.

5. Results and Analysis:

First investigate the accuracy of our scheme in pinpointing malicious service providers. Fig. 8a compares our scheme with the other alternative schemes (i.e., FTMV, PTMV, and RunTest) when malicious service providers aggressively attack different number of service functions. In this set of experiments, we have 10 service functions and 30 service providers. The number of service providers in each service function randomly ranges in [1, 8]. Each benign service provider provides two randomly selected service functions. The data rate of the input stream is 300 tuples per second. We set 20 percent of service providers as malicious. After the portal receives the processing result of a new data tuple, it randomly decides whether to perform data attestation. Each tuple has 0.2 probability of getting attested (i.e., attestation probability Pu ¼ 0:2), and two attestation data replicas are used (i.e., number of total data copies including the original data r ¼ 3). Each experiment is repeated three times. We report the average detection rate and false alarm rate achieved by different schemes. Note that RunTest can achieve the same detection accuracy results as the majority voting based schemes after the randomized probabilistic attestation covers all attested service providers and discovers the majority clique.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

In contrast, IntTest comprehensively examines both per-function consistency graphs and the global inconsistency graph to make the final pinpointing decision. We observe that IntTest can achieve much higher detection rate and lower false alarm rate than other alternatives. Moreover, IntTest can achieve better detection accuracy when malicious service providers attack more functions. And also observe that when malicious service providers attack aggressively, our scheme can detect them even though they attack a low percentage of service functions .

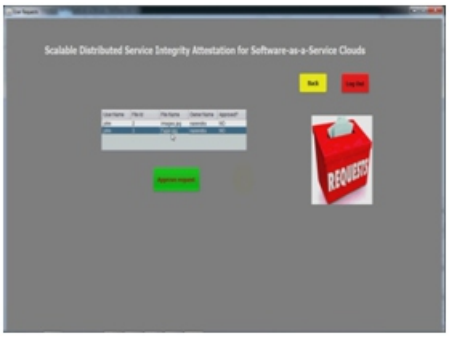

Figure 1

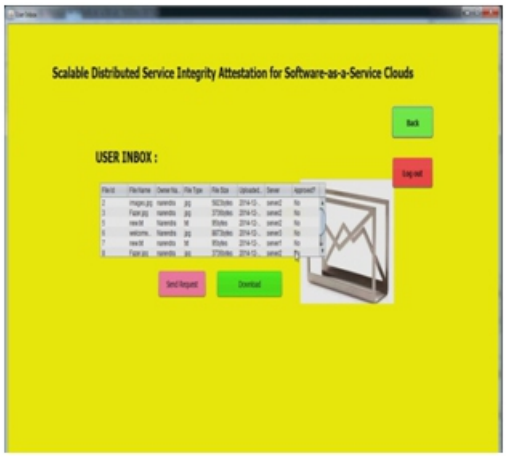

Figure 2

6 .CONCLUSION:

The design and implementation of IntTest, a novel integrated service integrity attestation framework for multitenant software-as-a-service cloud systems. IntTest employs randomized replay-based consistency check to verify the integrity of distributed service components without imposing high overhead to the cloud infrastructure. IntTest performs integrated analysis over both consistency and inconsistency attestation graphs to pinpoint colluding attackers more efficiently thanexisting techniques.

Furthermore, IntTest provides result autocorrection to automatically correct compromised results to improve the result quality. Also implemented IntTest and tested it on a commercial data stream processing platform running inside a production virtualized cloud computing infrastructure. This experimental results show that IntTest can achieve higher pinpointing accuracy than existing alternative schemes. IntTest is lightweight, which imposes low-performance impact to the data processing services running inside the cloud computing infrastructure.

REFERENCES:

[1]Amazon Web Services, http://aws.amazon.com/, 2013.

[2]Google App Engine, http://code.google.com/appen¬gine/, 2013.

[3] Software as a Service, http://en.wikipedia.org/wiki/Software asa Service, 2013.

[4] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, Web ServicesConcepts, Architectures and Applications (Data-Centric Systems andApplications). Addison-Wesley Pro¬fessional, 2002.

[5]T. Erl, Service-Oriented Architecture (SOA): Con¬cepts, Technology, andDesign. Prentice Hall, 2005.

[6] T.S. Group, "STREAM: The Stanford Stream Data Manager," IEEEData Eng. Bull., vol. 26, no. 1, pp. 19-26, Mar. 2003.

[7] D.J. Abadi et al., "The Design of the Borealis Stream Processing Engine," Proc. Second Biennial Conf. Innova¬tive Data Systems Research (CIDR '05), 2005.

[8] B. Gedik et al., "SPADE: The System S Declarative StreamProcessing Engine," Proc. ACM SIGMOD Int'l Conf. Managementof Data (SIGMOD '08), Apr. 2008.

[9] S. Berger et al., "TVDc: Managing Security in the Trusted VirtualDatacenter," ACM SIGOPS Operating Systems Rev., vol. 42, no. 1,pp. 40-47, 2008.

[10] T. Ristenpart, E. Tromer, H. Shacham, and S. Sav¬age, "Hey, YouGet Off My Cloud! Exploring Informa¬tion Leakage in Third-Party Compute Clouds," Proc. 16th ACM Conf. Computer and Communications-Security (CCS), 2009.