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

Versatile Distributed Service Integrity Check for Software-as-a-Service in Clouds

Sankranthi Mohan
M.Tech Student,
Department of Computer Science and Engineering,
Audisankara Institute of Technology, Gudur.

E.Ramesh Reddy, M.Tech
Assistant professor,
Department of Computer Science and Engineering,
Audisankara Institute of Technology, Gudur.

Abstract:

SaaS, or Software as a Service, describes any cloud service where consumers can able to access the relavent software applications over the internet. The applications are hosted in “the cloud” and can be used for a wide range of tasks for both individuals and organisations. Software-as-a service (SaaS) makes use software can updated them based on the services. In this paper, here we prapose IntTest, an effective service integrity attestation framework for SaaS clouds. IntTest provides an integrated graph attestation for analysis method that can pinpoint malicious service providers than existing methods. Also IntTest will get automatically and correct the corrupted results that are produced by the malicious service providers and replace them with best results will be produced by beginer service providers may get lead role.

Keywords: Service Integrity Attestation, Cloud Calculating, SaaS, Attestation,integrity software attestation.

Introduction:

Cloud computing is the lease of the same resources through which the users can use the resources depending upon the requirement and pay-off depend on the usage. Trough out cloud computing is the user can decrease the cost and effective can use the resource at any time. Data providers as a service providers are emerging to aggregate and manage such large data sets from different types of multiple sources to make this information more than easily available and usable to purpose of the businesses. These services will be used effectively for data rich, efficiency gains and process refinement.
There are three types of cloud as shown in fig1.

i)open cloud
ii)secret cloud
iii)mixed cloud

open cloud:

open cloud or external cloud is one in which the resources are leased on self service basis over the internet, via web applications/web services, third-party provider who shares the resources and bills on a fine-grained utility for computing basis. There is no authority to access about cloud in the cloud server. There is no accessability for applying software to the cloud service.

secret cloud:

secret cloud or internal cloud is used to describe the offerings that can be of private network. the clouds may be different approach for allocating the servers to the storages. The maximum clouds can be easily configurable and connectable servers can maintain the cloud environment and deploy in the environmenet. every one have private authorization to accessing the cloud services. the secret clouds not accessible to outside of cloud.

mixed cloud:

mixed cloud is one which contains multiple internal or external clouds that are public. AMES is based on platform as a service. Platform as a service (PaaS) is a category of cloud computing services that provides a basic environment a computing platform and a solution stack as a service.provisioning the capability of the software as services to be modified.The software can’t be modified once it has been stated as service.The challenge is managing multiple cloud providers and getting a diverse set of answers such as CRM, B2B, EAI, accounting and more to work together and Figure 2 shows the architecture of a typical cloud at a high level. An end user Bob connects to the big cloud via a portal from his browser. a set of cloud servers based on which the VM images can be run as a part, and optionally a storage pool to store persistent user data. the above IAAs and Paas,SaaS are different than the others. Social networking, cloud services and mobile touch points have turned business-to-consumer (B2C) on its ear. This trend requires some thought and consideration as we begin to see more importance on the consumer as a service (CaaS) platform. E -Commerce APIs that enable that contextual commerce experiences across touch points and that are now a new focus, layering on and directly impacting existing B2B platforms. Enterprise users are able to use the applications for a range of needs, including accounting and invoicing, tracking sales, that are planned and tuning, performance monitoring and communications (including webmail and instant messaging).SaaS is often to be referred to as a software for -on-demand and utilise them to taken to renting software rather gets than buying it. With the traditional software applications that you would purchase to the software upfront as a package and then install it onto your own computer and other Applications are purchased and used online with files saved in the cloud. computers may not understand the service which made by software.Almost everything everyone does today in the cloud as it pertains to data integration has been primarily business-to-business- (B2B-) focused. B2B in the cloud is now considered "business as usual" and is incrementally regarded as a mature technology for solving B2B issues because of focussing the software vendors are now shifting their focus into leveraging enterprise application integration (EAI) to facilitate data integration with database management systems in the cloud. This includes using cloud-based data integration and data management for integrating on-premise applications with each other, as well as integrating software as a service (SaaS) and PaaS and cloud applications with on-premise and/or other cloud-based applications.There are a fewer number of reasons to say that SaaS is very advantageous to the organizations and personal users alike:

1.don't required additional hardware costs; the processing power required to run the platform and applications is supplied ,managed and delivered by the cloud providers.
2.Updates are automated; whenever there is an update it is available into online to existing customers, often free of charge. No other things are required to add new software will be required as it often with any other types of uses that needed and the updates will be comes usually be deployed automatically done and verified by the cloud provider.

No other things are required to add new software will be required as it often with any other types of uses that needed and the updates will be comes usually be deployed automatically done and verified by the cloud provider.

6.Cross device compatibility; software applications can be accessed via any internet enabled device, which makes it ideal for those who can   uses a number of different devices, such different internet enabled devices and tablets, communication devices and those who don't always use the same computer but differs..

EXISTING SYSTEM:

Which enable application service providers (ASPs) to deliver their important applications via the massive cloud computing infrastructure. In particular,there may be our work focuses on data stream processing services that are considered to be one class of killer different applications. or example, there may be attackers can pretend to be estimate service providers to provide duplicate service components, and the service components provided by benign service providers may be include security holes that can be exploited by attackers. In the past scenarios there is no one related to act as a part of the service related to software hence the services are not comfort with the cloud.

DISADVANTAGES OF EXISTING SYSTEM:

•Which makes them difficult to be deploying on large-scale cloud computing infrastructures.
•They make very complexity of accessing the server in the cloud.
•There is no efficiency of accessing the services which provided inthe cloud .

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. in the proposed one there is a hierarchy fallowed on cloud to access the different types of services in the cloud. clouds are the integral part of servers nothing more than that in line.

ADVANTAGES OF PROPOSED SYSTEM:

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

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

•Here Both software act as a service oriented and configurable oriented into the cloud environmenent..

•In the cloud there may be a part of integrating the service as a software because of the dual cloud

DESIGN AND ALGORITHMS:

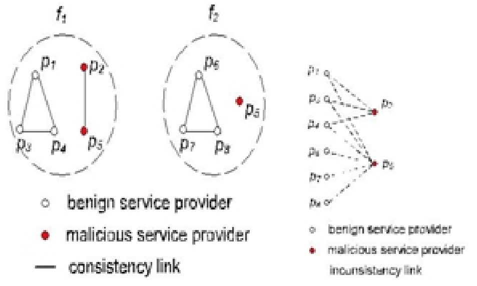
In this section, we first present the basis of the IntTest system: probabilistic replay-based consistency check and the integrity attestation graph model. We then describe the integrated service integrity attestation scheme in detail. Next, we present the result autocorrection scheme.

Baseline Attestation Scheme :

To detect service integrity attack and pinpoint malicious service providers, our algorithm relies on replay-based consistency check to derive the consistency/inconsistency relationships between service providers. For example, Fig. 2 shows the consistency check scheme for attesting three service providers p1, p2, and p3 that offer the same service function f. The portal sends the original input data d1 to p1 and gets back the result fðd1Þ. Next, the portal sends d01, a duplicate of d1 to p3 and gets back the result fðd01Þ. The portal then compares fðd1Þ and fðd01Þ to see whether p1 and p3 are consistent.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 exactly the same results. For example, the credit scores for the same person may vary by a small difference when obtained from different credit bureaus. We allow the user to define a distance function to quantify the biggest tolerable result difference.

Definition 1. For two output results, r1 and r2, which come from two functionally equivalent service providers, respectively, result consistency is defined as either r1 ¼ r2, or the distance between r1 and r2 according to user-defined distance function Dðr1; r2Þ falls within a threshold _.For scalability, we propose randomized probabilistic attes-tation, an attestation technique that randomly replays a subset of input data for attestation. For composite data-flow processing services consisting of multiple service hops, each service hop is composed of a set of functionally equivalent service providers. Specifically, for an incoming tuple di, the portal may decide to perform integrity attestation with probability pu. If the portal decides to perform attestation on di, the portal first sends di to a pre-defined service pathp1 ! p2 _ _ _ ! pl providing functions f1 ! f2 _ _ _ ! fl. After receiving the processing result for di, the portal replays theduplicate(s) of di on alternative service path(s) such as p01 ! p02 _ _ _ ! p0l, where p0j provides the same function fj as pj. The portal may perform data replay on multiple service providers to perform concurrent attestation.With replay-based consistency check, we can test func-tionally equivalent service providers and obtain their consistency and inconsistency relationships. We employ both the consistency graph and inconsistency graph to aggregate pairwise attestation results for further analysis. The graphs reflect consistency/inconsistency relationships across multiple service providers over a period of time. Before introducing the attestation graphs, we first define consis-tency links and inconsistency links.



(a) Per-function consistency graphs. (b) Global inconsistency graph.

Fig3:Attestation graphs.

Definition 2. A consistency link exists between two service providers who always give consistent output for the same input data during attestation. An inconsistency link exists between two service providers who give at least one inconsistent output for the same input data during attestation.

We then construct consistency graphs for each function to capture consistency relationships among the service providers provisioning the same function. Fig. 3a shows the consistency graphs for two functions. Note that two service providers that are consistent for one function are not necessarily consistent for another function. This is the reason why we confine consistency graphs within individual functions.

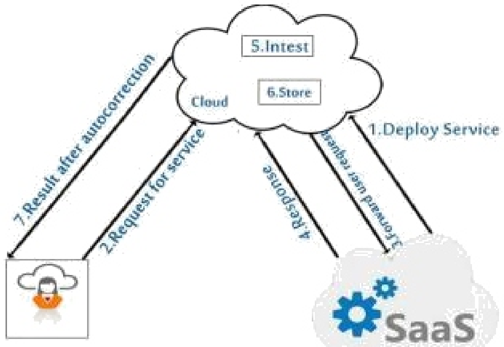
Definition 3. A per-function consistency graph is an undirected graph, with all the attested service providers that provide the same service function as the vertices and consistency links as the edges.

We use a global inconsistency graph to capture inconsistency relationships among all service providers. Two service providers are said to be inconsistent as long as they disagree in any function. Thus, we can derive more comprehensive inconsistency relationships by integrating inconsistency links across functions.

Fig. 3b shows an example of the global inconsistency graph. Note that service provider p5 provides both functions f1 and f2. In the inconsistency graph, there is a single node p5 with its links reflecting inconsistency relationships in both functions f1 and f2.

Definition 4. The global inconsistency graph is an undirected graph, with all the attested service providers in the system as the vertex set and inconsistency links as the edges.

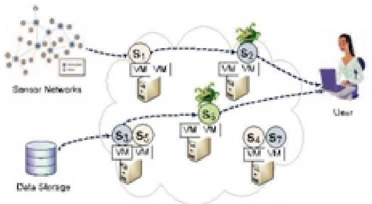
The portal node is responsible for constructing and maintaining both per-function consistency graphs and the global inconsistency graph. To generate these graphs, the portal maintains counters for the number of consistency results and counters for the total number of attestation data between each pair of service providers.



Over all architecture of the proposed method

Integrated Attestation Analysis:

Here we present an integrated attestation graph analysis algorithm. Step 1: continuity analysis: In the first step it will examine the per-function consistency graph and will pinpoint to The colluding attackers can try to escape from being it detected. Then next we must examine the perfection in consistency graph too. software may not configurable it may be usable.



Fiig 1:Service integrity attack in cloud-based data processing.

Step 2: Inconsistency analysis: This inconsistency of the graph will contain only the inconsistency links, this may exist in different possible combinations of the begining node and the malicious node set. then we can pinpoint a set of malicious service providers. If the two service providers are connected by automatically an inconsistency link, we can say that any one of them is malicious.

Conclusion:

In this paper we have introduced a different novel integrated services IntTest uses a reply based consistency check to verify the different service providers. IntTest will analyzes both the consistency and inconsistency showing graphs to find the malicious attackers efficiently than any other existing techniques. In future the cloud comes into the lead environment into the data warehouse architecture takes place .service providers are the cloud providers for the environment.Completely the data can be verified and integrated in the cloud part of the environment.

References:

[1]Juan Du, Daniel J. Dean, Yongmin Tan, Xiaohui Gu, 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]Q. Zhang, L. Cheng, and R. Boutaba, "Cloud Computing: State-ofthe-art and Research Challenges," in Journal of Internet Services and Applications, vol. 1, no. 1, pp. 7–18, Apr. 2010.

[3]Shi.E, Perrig.A, and Doorn.L.V, "Bind: A fine-grained attestation service for to secure distributed systems," in Proceedings of the IEEE Symposium on Security and Privacy, 2005.

[4]B. Aggarwal, N. Spring, and A. Schulman, "Stratus : EnergyEfficient Mobile Communication using Cloud Support," in ACM SIGCOMM DEMO, 2010.

[5] T. Erl, Service-Oriented Architecture (SOA): Concepts, Technology, and Design. Prentice Hall, 2005.

[6] T.S. Group, "STREAM: The Stanford Stream Data Manager," IEEE Data 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. Innovative Data Systems Research (CIDR '05), 2005.

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

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

[10]T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, You Get Off My Cloud! Exploring Information Leakage in Third-Party Compute Clouds," Proc. 16th ACM Conf. Computer and Commu-nications Security (CCS), 2009.

[11]W. Xu, V.N. Venkatakrishnan, R. Sekar, and I.V. Ramakrishnan, "A Framework for Building Privacy-Conscious Composite Web Services," Proc. IEEE Int'l Conf. Web Services, pp. 655-662, Sept. 2006.

[12]P.C.K. Hung, E. Ferrari, and B. Carminati, "Towards Standardized Web Services Privacy Technologies," IEEE Int'l Conf. Web Services, pp.174-183, June 2004.

[13]L. Alchaal, V. Roca, and M. Habert, "Managing and Securing Web Services with VPNs," Proc. IEEE Int'l Conf. Web Services, pp. 236-243, June 2004.

[14]H. Zhang, M. Savoie, S. Campbell, S. Figuerola, G. von Bochmann, and B.S. Arnaud, "Service-Oriented Virtual Private Networks for Grid Applications," Proc. IEEE Int'l Conf. Web Services, pp. 944-951, July 2007.

[15]M. Burnside and A.D. Keromytis, "F3ildCrypt: End-to-End Protection of Sensitive Information in Web Services," Proc. 12th Int'l Conf. Information Security (ISC), pp. 491-506, 2009.

[16]I. Roy et al., "Airavat: Security and Privacy for MapReduce," Proc. Seventh USENIX Conf. Networked Systems Design and Implementation (NSDI), Apr. 2010.