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

Infrastructure Attacks Service Integrity Accuracy on Software – As - A Service

O.Saikrishnamaraju
M.Tech Student
Department of CSE
Visvodaya Engineering College
Kavali.

N.Srinadh Reddy
Associate Professor
Department of CSE
Visvodaya Engineering College
Kavali.

Abstract:
Software-as-an accommodation (SaaS) makes utilization of a cloud computing infrastructure to distribute their applications to many users regardless of their location. Because of this sharing nature SaaS clouds are vulnerably susceptible and provide more opportunities for assailants to exploit the system susceptibility and perform strategic attacks. In this paper, we present IntTest, an efficacious accommodation integrity attestation framework for SaaS clouds. IntTest provides an integrated graph attestation analysis method that can pinpoint malignant accommodation providers than subsisting methods. Additionally IntTest will automatically redress the corrupted result that are engendered by the malevolent accommodation providers and supersede it with good results engendered by benign accommodation providers. Our experimental results show that our scheme is efficacious and can achieve higher precision in pinpointing the assailants than the subsisting approaches.

Keywords: ***Software-as-a service, Infrastructure, attacks, Service Integrity, Accuracy.***

1. Introduction

Cloud computing relies on sharing of resources over a network. Cloud computing mainly fixates on maximizing the efficacy of the shared resources. Software as an accommodation describes any cloud accommodation where consumers are able to access software applications over the cyber world. Clouds are providing many types of accommodations like applications, infrastructures etc. Software-as-a-accommodation (SaaS) clouds (e.g., Amazon Web Accommodation (AWS) and Google AppEngine) build upon the concepts of software as an accommodation and serviceoriented architecture (SOA) which enable application accommodation providers (ASPs) to distribute their applications via the massive cloud computing infrastructure[1].Cloud computing infrastructures are shared by utilizing ASPs from different security domains, because of that its vulnerably susceptible.

Now a days the cloud computing technology is popular because it is a magnetizing technology in computer science field. This paper concentrate on the integrity attacks on software as an accommodation clouds and because of that the utilizer will receive deplorable results after processing the data. Fig.1 shows the integrity attacks in software as an accommodation clouds. Majority of software as an accommodation cloud solutions are predicated on a multitenant architecture. In the antecedent research papers confidentiality and privacy bulwark quandaries are studied extensively but the accommodation integrity attestation quandary was not opportunely addressed. In

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

software as an accommodation cloud one of the most consequential quandaries that need to be addressed is this accommodation integrity, no matter whether the data processing in cloud is public or private data. In the antecedent papers they are provided some software integrity attestation techniques but most of them requires special trusted hardware or secure kernel fortifies and because of these reasons that cannot be deployed in sizably voluminous scale cloud computing. This paper presents IntTest, an incipient framework for multi tenant cloud systems. This technique provides the novel integrated attestation graph analysis technique that will provide a more vigorous assailant pinpointing power than the subsisting schemes. It will automatically enhance the result quality by superseding the deplorable results that are engendered by the assailers by good results that are engendered by the benign accommodation providers. This can achieve higher assailant pinpointing precision than subsisting techniques Run Test and Adap Test.

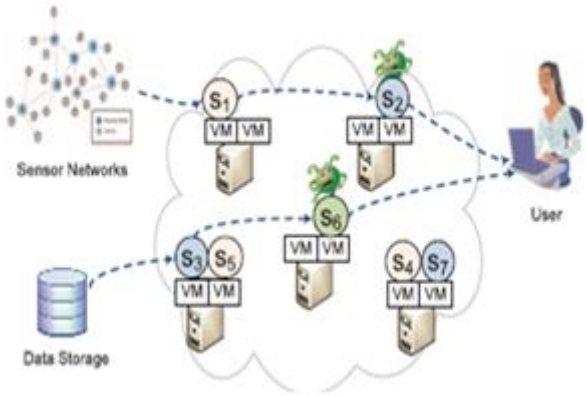

Fig 1: Service integrity attacks in clouds.

Concretely, RunTest and AdapTest as well as traditional majority voting schemes need to surmise that benign accommodation providers take majority in every accommodation function. In sizably voluminous-scale multitenant cloud systems, immensely colossal number of malignant assailers may launch colluding attacks on the targeted accommodation functions to make them malevolent. To address this challenge, IntTest takes a holistic approach by systematically examining both consistency and erraticism relationships among different accommodation providers within the entire cloud system. IntTest checks both per-function consistency and the ecumenical erraticism graphs. An advantage of utilizing this IntTest is it cannot only pinpointing the malignant assailants more efficiently but additionally it can suppress truculent assailers and withal limit the scope of damage that are caused by the assailments. The experimental result shows that IntTest can achieve more precision in pinpointing malignant assailers than any other subsisting schemes. Additionally this IntTest is more scalable and it will reduce overhead engendered by the attestation more than the other voting schemes.

This paper implements Efficient and distributed service integrity attestation

- Efficient and distributed service integrity attestation framework for large scale cloud computing infrastructures.
- An integrated service integrity attestation scheme that can achieve higher pinpointing accuracy than existing techniques.
- A result auto correction technique is used that will• automatically correct the corrupted results produced by malicious attackers and replace it with good results.
- The analytical study and experimental evaluation used to quantify the accuracy and overhead of the service integrity attestation method.

The rest of this paper is organized as follows. Section 2 presents the cognate work. Section 3 provides the proposed system in detail. Section 4 presents the main modules. Determinately, the paper concludes in section 5.

2. Related Work

In recent years many integrity attestation schemes have been developed for software as an accommodation clouds. For example the BIND technique, AdapTest technique, RunTest technique etc. but all of these are having some quandaries some of them needs secure kernel support and special trusted hardware

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

components. In BIND (Binding Information and Data) technique is a verification method of integrity accommodations that are provided by the software as an accommodation 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 utilizes the following steps:
1) Attestation annotation mechanism
2) Sandbox mechanism
3) Verification of authenticator through hash.

BIND method utilizes the DiffeeHellman key exchange for the purport of integrity attestation. Another subsisting technique is TEAS (Timed Executable Agent System) this is utilized for forfending the integrity of cloud computing platforms. An agent generation and verification algorithm is utilized in this TEAS method. Another one subsisting 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 mendacious data flow processing and will pinpoint mallicious data processing accommodation provider and atlast it will detect the assailers deportment. This RunTest will provide the benign accommodation providers and will determine the malevolent deportment of the assailants. But the disadvantage is its low performance.

The AdapTet is another one subsisting technique, it provides a novel adaptive data driven runtime accommodation integrity attestation framework. This method will significantly reduce the overhead of attestation and will shorten the delay. It treats all components as ebony boxes and it does not require any special hardware or software requisites. In this AdapTest it will reduce the attestation overhead and the detection of maleficent assailers or accommodation providers will be high when compared to other techniques. All the above methods that are utilized in the subsisting papers are having some disadvantages. And to surmount that disadvantages this IntTest is utilizing. And by utilizing this IntTest it will provides more integrity and it will provide more precision in pinpointing the malignant assailers and accommodation providers. Withal it will provide a result auto rectification method and will redress the deplorable results and supersede it with good results and additionally in this it does not require any special hardware and secure kernel support.

Proposed System:

Software as an accommodation and accommodation oriented architecture are the fundamental concepts of SaaSclouds and this will sanction the application accommodation provider to distribute their application via cloud computing infrastructure. In our proposed method we are introducing an incipient concept called IntTest. The main goal of IntTest is, it can pinpoint all the malevolent accommodation providers. IntTest will treat all the accommodation providers as ebony boxes and this does not require any special hardware or secure kernel support. When we are considering the astronomically immense scale cloud system multiple accommodation providers may simultaneously compromised by a single malevolent assailant. In this we surmise that the malevolent nodes are not having any erudition about the other nodes except those which they are directly interacting. In this proposed system we are making some posits.

First of all we are postulating that the total number malignant accommodation components are less than that of the total number of benign accommodation providers in the entire cloud. This posits is very paramount because without this postulation, it would be arduous for any assailment detecting scheme to work successfully. The second posit is the data processing accommodations are consequential deterministic. That is, the same input that is giving by a benign accommodation component will always engender the same output. And conclusively we surmise that the erraticism caused by hardware or software faults can be omitted from malevolent

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

attacks. Fig. 2 shows the overall architecture of the proposed system. In this the utilize give request to cloud the serve will be deployed in the cloud the cloud will forward the utilizer request to the SaaS and the replication will be send to the cloud by the SaaS. And then the IntTest process will be done. After that the result auto rectification will be done. After that the result will be send to the utilize by the cloud. The architecture shows this IntTest module in detail

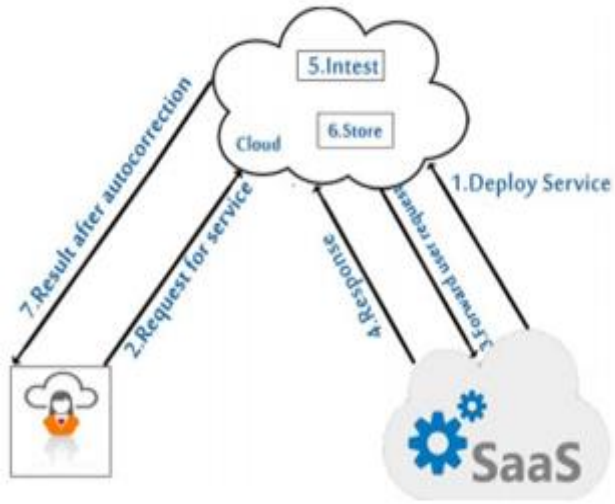

Fig 2: Over all architecture of the proposed method.

3. Implementation

3.1 Baseline Attestation Scheme

IntTest is utilized to detect the accommodation integrity attack and to pinpoint maleficent accommodation providers. For that first we are deriving the consistency and erraticism relationship between accommodation providers. Instead, we replay the attestation data on different accommodation providers after receiving the processing result of the pristine data. Thus, the malignant assailers cannot evade the jeopardy of being detected when they engender mendacious results on the pristine data. Albeit the replay scheme may cause delay in a single tuple processing, we can overlap the attestation and mundane processing of consecutive tuples in the data stream to obnubilate the attestation delay from the utilizer.

3.2 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 accommodation providers. The consistency links in the consistency graph will provide a set of accommodation providers. It will keep consistent with each other on a categorical accommodation function. The benign accommodation providers will always keep consistent with each other and will compose a clique in terms of consistency links. The colluding assailants can endeavor to elude from being detected.

Then next we must examine the perfunction in consistency graph additionally. Step 2: Erraticism analysis: This erraticism graph will contain only the erraticism links, this may subsist in different possible cumulations of the benign node and the maleficent node set. First we postulate that the total number of malignant accommodation providers in the cloud system is not more than the benign accommodation providers, then we can pinpoint a set of malignant accommodation providers. If two accommodation providers are connected by an erraticism link, we can verbalize that any one of them is maleficent.

3.3 Result Auto Correction for Attacks

IntTest can not only pinpoint malevolent accommodation providers but additionally it will autocorrect the corrupted data processing results with good results to amend the result quality of the cloud data processing accommodation. Without our attestation scheme, once if a pristine data input is transmuted by any maleficent assailant, then the processing result of that input will be corrupted and which will result in degraded result quality. IntTest provides the attestation data and the malevolent node pinpointing results to detect and rectify compromised data processing results[1]. IntTest will examine both the erraticism and consistency graphs to make a final decision to pinpoint the maleficent accommodation provider. This technique can achieve higher detection

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

rate than any other subsisting technique and will have low erroneous alarm rate than others. Additionally IntTest can achieve higher detection precision than any other techniques when malevolent accommodation providers attack more nodes. This method will identify the assailers albeit they assail a very low percentage of accommodations.

4. Conclusion

In this paper we introduced a novel integrated accommodation integrity attestation graph analysis scheme for multitenant software-as-a-accommodation cloud system. IntTest utilizes a replication predicated consistency check to verify the accommodation providers. IntTest will analyses both the consistency and erraticism graphs to find the malignant assailers efficiently than any other subsisting techniques. And additionally it will provide a result auto rectification to amend the result quality.

5. 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] Du.J, Wei.W, Gu.X, and Yu.T, "Runtest: Assuring Integrity of Dataflow Processing in Cloud Computing Infrastructures," Proc.ACM Symp. 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.

[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 Communications 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.

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

[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.

Author Details

O.Saikrishnamaraju received B.Tech degree in Information Technology from Narayana Engineering College, JNTUA, Anantapur, A.P, and India in 2012 and Pursuing M.Tech in Computer Science and Engineering in Visvodaya Engineering College Kavali from JNTUA Ananthapur, A.P.

N.Srinadh Reddy received M.Tech in 2006 Computer Science and Engineering A.P, India. Working as Associate Professor in the department of CSE in Visvodaya Engineering College in Kavali having 13 years Experience in lecturer field.