IJETIE International Journal of Emerging Technology and Innovative Engineering
Volume I, Issue 3, March 2015
ISSN: 2394 - 6598
www.ijetie.org

ASSURING INTEGRITY OF DATAFLOW PROCESSING IN CLOUD COMPUTING INFRASTRUCTURE

R.Genga devi[1], K.Anitha[2], M.Murugeshwari[3],S.vidhya[4], Dr.K.Ramasamy[5]

1, 2, 3- UG STUDENT, P.S.R.RENGASAMY COLLEGE OF ENGINEERING FOR WOMEN ,SIVAKASI

4, - ASSISTANT PROFESSOR,, P.S.R.RENGASAMY COLLEGE OF ENGINEERING FOR WOMEN, SIVAKASI

5- PRINCIPAL, P.S.R.RENGASAMY COLLEGE OF ENGINEERING FOR WOMEN, SIVAKASI
[1]rgenga11@gmail.com, [2]rrajam79@gmail.com, [3]murugeshwari03@gmail.com

ABSTRACT:

In cloud service provider will provides different types of service. In our project they are using Software as a Service (**SaaS).** In the cloud service provider will enables application will be shared via massive cloud infrastructure. In the type of sharing nature they will spread the infected code and then make the system more vulnerable then perform malicious attack in the service provider. In the project, they will perform intruder test and make the system is scalable and effective service integrity in the Software as a Service Provider (**SaaS).** In intruder test can provides the novel integrated attestation scheme provide stronger attacker in the pin point scheme in the previous models. In the Intruder test will automatically provides good results when will provide a better when compared to previous of malicious attack code. So, it is very effective for the cloud shares. In the type will provide secure to all type of files. In the model will provide effective result in the type of intruder test. So, they will share effective information in the Software as a Service Provider (**SaaS).** Therefore, any type of malicious attacker will not attack our files. So, the Intruder test will be used in the project.

Keywords:

Intruder test, Malicious attacker, Pin point power, Software as a Service, Integrated attestation scheme.

Introduction:

Cloud computing is a emerged technique because of the physical system infrastructure will not have enough memory space. So, we are using the clod computing. In the Google App Engine will also in built with the Software as a Service (**SaaS),** Service Oriented Architecture (**SOA**) and enables the application service providers (**ASPs**) to deliver their applications via the massive cloud computing infrastructure. In the project will store the massive amount of data stream and then they will uses Intruder Test. In the Traditional system they use the Byzantine fault tolerance (**BFT**) technique and pin point power technique and then our system uses the Intruder test. Intruder test examines the both per-function consistency graphs and the global inconsistency graph. In per-function consistency graph will shows the analysis and scope of the colluding attackers. In the global inconsistency graph will shows to expose the attackers to compromise the system. In Intruder test will provides auto correction of the data stream

IJETIE
International Journal of Emerging Technology and Innovative Engineering
Volume I, Issue 3, March 2015
ISSN: 2394 - 6598
www.ijetie.org

in the system. So, they will first using the Intruder test and the auto correction model for Software as a Service provider model (**Saas**).

System Design:

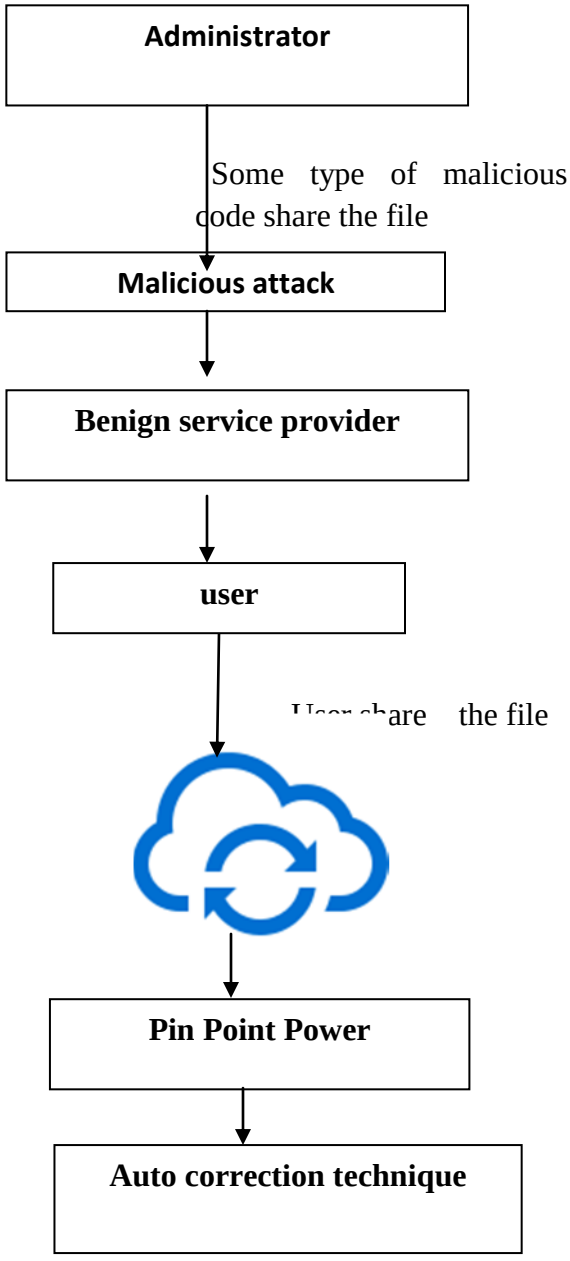
In this paper we are using the Intruder test technique and auto correction techniques. We make the following contributions.

- We propose an Intruder test technique, what type of data stream will be attacked and then test the data stream.
- We also propose a auto correction technique it will be used to automatically correct the intruder information.
- We carry out the technique to provide more security and then enhance a system model.

In the **admin phase,** they will share the file in the cloud and the admin will also checks any type of malicious code will be found in the file. The admin will use Intruder test and then they manage these type of malicious file. In the admin will also uses the auto correction technique to correct the uploading the file. So, they will manage only correct type of file to upload it.

In the **Malicious attack phase,** will any type of malicious code in the cloud share. In the clod these type any vulnerability of attack will be find out and manage in the admin phase. So, they we are using the Intruder test and auto correction technique.

In the **Benign service provider,** is nothing but checks the any type of attack in the system. In the type of service provider will checks the file and upload to cloud. So, the malicious code will not share the system.



IJETIE International Journal of Emerging Technology and Innovative Engineering
Volume I, Issue 3, March 2015
ISSN: 2394 - 6598
www.ijetie.org

In the **user phase,** the user will wants to share the data stream into the cloud. In the type of sharing any type of vulnerability will proof into the system. In the System, we are using some techniques to remove these vulnerabilities.

In the **Pin point power,** is the traditional technique for remove the vulnerability. In the approach we are remove small amount of attack. But, those type of system using the Intruder test to remove the malicious code. It will be used to any kind of attack to remove it.

In the **Auto correction Technique,** will be used to automatically correct that attack. In the phase, any type of viruses and vulnerability can be removed. So, it will produce very effective sharing.

Implementing Safe Shares:

A Safe share will provides some of the technique. These technique are used to satisfy and correct the malicious code in the attack.

Pin Point Power Algorithm:

In the system will also uses this type of algorithm. And then also provides very safe share of file or data stream in the cloud.

Algorithm:

1 . for every M € [|CG|, ⌊N/2⌋]
2. Ω = Ø, R = Ø
3. for every node p in G
4. compute |Np| + |CG′|
5. if (|Np| + |CG′| > M)
6. Ω= Ω ∪ {p}
7. final malicious node set R = R ∪ Ω
8. if R = Ø
9. continue
10. else
11. for every Gi
12. compute Mi
13. set Ωi to the subset of appearing in the Gi
14. if (Ωi ∩Mi ≠ Ø)
15. R = R ∪Mi
16. return all sets of R

Simulation Parameters:

There are different type of simulation parameters are used to detect the system performance. In the type we are use no of users in the cloud and then the how many data streams are uploaded in the cloud. And how many minutes to take to complete the process. In the process there are two graphs will be used.

Graph Analysis Time for

Intruder Test

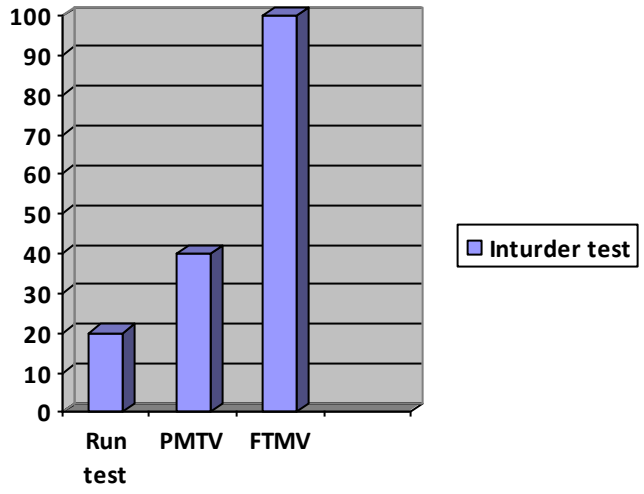
No.of Providers	Consistency graph	Inconsistency graph
200	4.22 ± 0.018 ms	1.64 ± 0.001 ms
400	15.89± 0.013ms	6.52 ± 0.004 ms
600	35.29 ± 0.095 ms	15.11 ± 0.015 ms

IJETIE
International Journal of Emerging Technology and Innovative Engineering
Volume I, Issue 3, March 2015
ISSN: 2394 - 6598
www.ijetie.org

800	**62.62 ± 0.021 ms**	**26.18 ± 0.350 ms**
1000	**100.00 ± 0.434 ms**	**39.98 ± 0.179 ms**
1200	**120.00 ± 0.544 ms**	**49.98 ± 0.119 ms**

Using that type of the graph analysis, we have analysis the time of the security share. We are also use the auto correction method to deploy for the security file share.

ATTESTATED OVERHEAD COMPARISION



Related Work:

Distributed Authorization: If a dynamic cloud Environment, the consistency of distributed environment for security shares. It describes an inconsistency issue that can be arise in the authorization policies are in a static but it can be revoked or altered. These inconsistency leads to an unsafe share of the data stream.

Relaxed Consistency Model: in a cloud environment the database solution can be written. In large scale applications it will introduce a new set of consistency problems when it will provided the new dimensions.

In the Intruder test will also provide the attested use of the data stream and then how to use of it. In the auto correction technique is used to correct that type of malicious file. In the related system will sometimes makes inconsistency and dynamic environment.

Future work:

In the Paper, will also provides some new techniques to test any type of file. In they will also uses dynamic, scalable environment in future. In the paper will also exhibits in any size of file will used to share. Some new techniques are added to the system and it will make the system very fast. In the System will check any type of redundant malicious code in the system it will be easily find out.

So, it will uses any type of auto correction technique is used to correct any type of file (i.e., video, audio, image an text file also)

Conclusion:

In this paper, we have presented the design and implementation of Intruder Test, and a novel integrated service attestation framework for multi-tenant software-as-a-service (**Saas)** cloud systems. Intruder Test enables randomized replay-based consistency check and verify the integrity of distributed service components without imposing high overhead of the cloud infrastructure. Intruder Test also performs integrated analysis over both consistency and global inconsistency attestation graphs to pinpoint colluding attackers more efficiently than existing techniques. Furthermore, Intruder Test provides result to auto correction to automatically correct compromised results to improve the result quality. We have implemented Intruder Test and tested it on a commercial stream of data processing platform running inside a production virtualized cloud computing infrastructure.Our experimental results will show that Intruder Test can achieve higher pinpointing accuracy than existing alternative schemes. Intruder Test is light-weight, which imposes low performance
impact to the input data processing, services running inside the cloud computing. It will also detect any type of malicious code present in the shared file.

REFERENCES:

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

[2] "Google App Engine," http://code.google.com/appengine/.

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

[4] G. A. amd F. Casati, H. Kuno, and V. Machiraju, "Web Services
Concepts, Architectures and Applications Series: Data-Centric Systems
and Applications," *Addison-Wesley Professional*, 2002.

[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 Engineering Bulletin, 26(1):19-26*, Mar. 2003.

[7] D. J. Abadi and et al, "The Design of the Borealis Stream Processing
Engine," *Proc. of CIDR*, 2005.

[8] B. Gedik, H. Andrade, and et. al., "SPADE: the System S Declarative
Stream Processing Engine," *Proc. of SIGMOD*, Apr. 2008.

[9] S. Berger, R. Caceres, and et. al., "TVDc: Managing security in the
trusted virtual datacenter," *ACM SIGOPS Operating Systems Review*,
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," in *Proceedings of the 16th ACM Conference on Computer
and Communications Security (CCS)*, 2009.

[11] J. Garay and L. Huelsbergen, "Software integrity protection using timed executable agents," in *Proceedings of ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, Taiwan, Mar. 2006.

[12] T. Garfinkel, B. Pfaff, and et. al., "Terra: A virtual machine-based platform for trusted computing," in *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP)*, Oct. 2003.

[13] R. Sailer, X. Zhang, T. Jaeger, and L. van Doorn, "Design and implementation of a tcg-based integrity measurement architecture," in *Proceedings of 13th USENIX Security Symposium*, San Diego, CA, Aug. 2004.

[14] A. Seshadri, M. Luk, E. Shi, A. Perrig, L. van Doorn, and P. Khosla, "Pioneer: Verifying code integrity and enforcing untampered code execution on legacy systems," in *Proceedings of the 20th ACM Symposium on Operating Systems Principles (SOSP)*, Oct. 2005.

[15] A. Seshadri, A. Perrig, L. V. Doorn, and P. Khosla, "Swatt: Softwarebased attestation for embedded devices," in *IEEE Symposium on Security and Privacy*, May 2004.

[16] S. Berger, R. Caceres, and et. al., "TVDc: Managing security in the trusted virtual datacenter," *ACM SIGOPS Operating Systems Review*, vol. 42, no. 1, pp. 40–47, 2008.

[17] E. Kaiser, W. Feng, and T. Schluessler, "Fides: Remote anomalybased cheat detection using client emulation," in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS)*, 2009.

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

[19] F. Monrose, P. Wyckoff, and A. D. Rubin, "Distributed execution with remote audit," in *Proceedings of ISOC Network and Distributed System Security Symposium (NDSS)*, Feb. 1999.

[20] M. Alam, M. Nauman, X. Zhang, T. Ali, and P. C. Hung, "Behavioralattestation for business processes," in *IEEE International Conference*