Scalable Distributed Service Integrity Attestation for Software as a Service Clouds

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.

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.

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.

ADVANTAGES OF PROPOSED SYSTEM:

- A scalable and efficient distributed service integrity attestation framework for large scale cloud computing infrastructures.
- A novel integrated service integrity attestation scheme that can achieve higher pinpointing accuracy than previous techniques.
- A result auto correction technique that can automatically correct the corrupted results produced by malicious attackers.
- Both analytical study and experimental evaluation to quantify the accuracy and overhead of the integrated service integrity attestation scheme.

MODULES:

1. Baseline Attestation
2. Integrated Attestation
   a. Consistency graph analysis
   b. Inconsistency graph analysis
3. Auto correction For Attacks

MODULES DESCRIPTION:

Baseline Attestation:

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 an ingle 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 relation-ship 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 services providers may produce similar but not exactly the same results.

Integrated Attestation:

A. Consistency graph analysis: We first examine per-function consistency graphs to pinpoint suspicious service providers. The consistency links in per-function 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. However, strategically colluding attackers can try to take majority in a specific service function to escape the detection. Thus, it is insufficient to examine the per-function consistency graph only. We need to integrate the consistency graph analysis with the inconsistency graph analysis to achieve more robust integrity attestation.

B. Inconsistency graph analysis: Given an inconsistency graph containing only the inconsistency links, there may exist different possible combinations of the benign node set and the malicious node set. However, if we assume that the total number of malicious service providers in the whole system is no more than K, we can pinpoint a subset of truly malicious service providers. Intuitively, given two service providers connected by an inconsistency link, we can say that at least one of them is malicious since any two benign service providers should always agree with each other. Thus, we can derive the lower bound about the number of malicious service providers by examining the minimum vertex cover of the inconsistency graph. The minimum vertex cover of a graph is a minimum set of vertices such that each edge of the graph is incident to at least one vertex in the set.

Auto correction For Attacks:

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, 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.

SYSTEM REQUIREMENTS:

HARDWARE REQUIREMENTS:

Ø System	:	Pentium IV 2.4 GHz.
Ø Hard Disk	:	40 GB.
Ø Floppy Drive	:	1.44 Mb.
Ø Monitor	:	15 VGA Colour.
Ø Mouse	:	Logitech.
Ø Ram	:	512 Mb.

SOFTWARE REQUIREMENTS:

Ø Operating system	:	Windows XP/7.
Ø Coding Language	:	JAVA/J2EE
Ø IDE	:	Netbeans 7.4
Ø Database	:	MYSQL