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

A Nice Intrusion Detection and Defence System in Software as a Service

M.Lakshmi Priyanka
M.Tech,
Audisankara College of Engineering and Technology, Gudur.

B. Sudhakar
Associate Professor,
Audisankara College of Engineering and Technology, Gudur.

ABSTRACT:

The cloud technologies have been making enormous progress in the field of providing software along with their environment as a service. The providers of the SaaS cloud have to invest on a massive cloud infrastructure. And those clouds have to be capable of sharing nature among them. This nature could cause the cloud to become vulnerable to malicious attacks. Here the main focus of our paper is to discuss the limitations of an existing system and providing a solution to overcome those limitations to enhance the performance of the cloud infrastructure. The main idea we present is based on detection of the vulnerable soft spots in the cloud environment and equip those with high level attack detection and defence systems. This system can effectively reduce the cost by prevention of attack than recovery after attack.

Index terms:

cloud, SaaS, masquerade, Int-Test, NICE, SAG, MFA, TFA.

1.INTRODUCTION:

The SaaS ability of cloud has been adopted through the modern organizational sector. They are indeed a special privileges of the multinational organisations like IBM with their owned clouds. These clouds give their consumer to minimize their software costs by providing better software services at low usage based payment. The main feature and loophole of the SaaS cloud is its nature to share the data in between the anonymous similar service vendors though internet.

Thus this cause a possibility for the hacker to pounce over a sector of customers for a particular type of services in the cloud as a credible vendor and misbehave with them to damage the reputation of the entire cloud.To avoid the above problem the previous studies provided a valiant enough effort to put together a system that uses the consistency check process to detect the masquerade in the group of the common data vendors. But this system has certain limitations which we are going to discuss in brief in the following sections.

2.EXISTING SYSTEM:

In this section, we first present the basis of the Int-Test 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 auto-correction scheme.

2.1. 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. 1 shows the consistency check scheme for attesting three service providers p1, p2, and p3 that offer the same servicefunctionf.

The portal sends the original input data d1 to p1and gets back the result f (d1). Next, the portal sends d1l, aduplicate of d1 to p3 and gets back the result f (d1l). Theportal then compares f (d1) and f (d1l) to see whether p1 andp3 are consistent.

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 = r2, or the distance between r1 and r2 according to user-defined distance function D (r1,r2) falls within a threshold α.

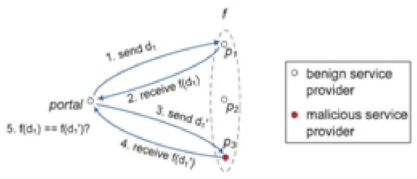


Fig. 1: Replay-based consistency check

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.

With replay-based consistency check, we can test functionally 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.

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.

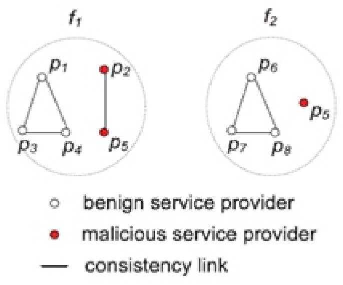


Fig. 2: per-function consistency graphs.

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.

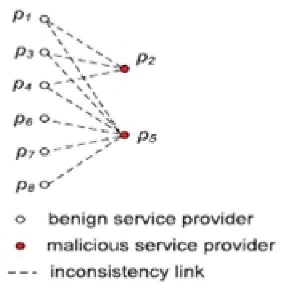


Fig. 3: Global inconsistency graph.

2. 2. Integrated Attestation Scheme

We now present our integrated attestation graph analysis algorithm.

Step 1: Consistency graph analysis. We first examine per-function consistency graphs to pinpoint suspicious service providers.

Step 2: 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.

We now define the residual inconsistency graph for a node pi as follows.

Definition 5. The residual inconsistency graph of node pi is the inconsistency graph after removing the node pi and all of links adjacent to pi.

Step 3: Combining consistency and inconsistency graph analysis results. Let Gibe the consistency graph generated for service function fi, and G be the global inconsistency graph.

2. 3.Limitation Discussion:

Although we have shown that Int-Test can achieve better scalability and higher detection accuracy than existing schemes, Int-Test still has a set of limitations that require further study. A detailed limitation discussion can be found in Section 4 of the online supplementary material. We now provide a summary of the limitations of our approach. First, malicious attackers can still escape the detection if they only attack a few service functions, take majority in all the compromised service functions, and have less inconsistency links than benign service providers. However, Int-Test can effectively limit the attack scope and make it difficult to attack popular service functions. Second, Int-Test needs to assume the attested services are input deterministic where benign services will return the same or similar results defined by a distance function for the same input. Thus, Int-Test cannot support those service functions whose results vary significantly based on some random numbers or time stamps.

2.PROPOSED SYSTEM:

In the proposed system we use the defence in depth mechanism of detecting the intrusion in the cloud vendors by using the NICE model. The NICE model identified the vulnerable areas in the cloud network.

3. 1. NICE Model:

The Material Intrusion perception and Countermeasures Election in realistic meshing systems (Metropolis) is old to plant a defence-in-depth intrusion find framework. It incorporates act represent analytical procedures into the intrusion reception processes.

A NICE-A periodically scans the virtual method vulnerabilities within a cloud server to give Scenario Onslaught Graph (SAGs), and then based on the stiffness of identified danger toward the collaborative crime goals. Overnice optimizes the exploit on cloud servers to lessen inventiveness t.b.. Prissy employs a new onslaught represent act for attack detection and bar by correlating fight activity and also suggests effectual countermeas

Multi-factor validation (also MFA, Two-factor mark, TFA, T-FA or 2FA) is an move to proof which requires the representation of two or statesman of the iii mark factors: a noesis integer (“something only the somebody knows”), a firmness figure (“something only the somebody has”), and an inherence constant (“something only the somebody is”).

After show, each reckon staleness be validated by another lot for marking to occur. Two-factor marking is commonly institute in electronic machine validation, where rudimentary substantiation is the knowledge of a requesting entity presenting several inform of its individuality to a position entity.

3. 2. User Request:

The considered individual record and login themselves. The user registration and login is through using multifactor marking. The multifactor authentication includes the memory of username and parole, onetime parole to user’s net-mail tact and written countersign mark. They content for the ingeniousness they essential in the cloud. This petition is monitored by Realistic machine Monitoring.

3. 3. Cloud server :

The person content in the virtual machine monitoring is analysed by the cloud server. The server greeting to the soul by retrieving the accumulation from the database.

3. 4. Virtual Machine Profiling:

The VM profiling analyses the wrongdoer in the Virtual organisation monitoring Grouping. Realistic machines in the darken can be profiled to get punctilious info around their nation, services flowing, give ports, and so on. One starring compute that counts toward a VM profile is its connectivity with added VMs. Any VM that is attached to solon signal of machines is much determinative than the one joined to fewer VMs because the meaning of compromise of a highly neighbouring VM can effort statesman alteration.Also required is the knowledge of services running on a VM so as to swear the credibleness of alerts pertaining to that VM. An attacker can use port-scanning syllabus to execute a consuming ports on any VM. So assemblage nigh any staring ports on a VM and the history of opened ports plays a big portrayal in determining how insecure the VM is. All these factors compounded leave structure the VM saliency:

```
Algorithm 1. Alert_Correlation
Require:   alert $a_c$, $SAG$, $ACG$
 1: if ($a_c$ is a new alert) then
 2:     create node $a_c$ in $ACG$
 3:     $n_1 \leftarrow v_c \in map(a_c)$
 4:     for all $n_2 \in parent(n_1)$ do
 5:         create edge $(n_2.alert, a_c)$
 6:         for all $S_i$ containing $a$ do
 7:             if $a$ is the last element in $S_i$ then
 8:                 append $a_c$ to $S_i$
 9:             else
10:                 create path $S_{i+1} = \{subset(S_i, a), a_c\}$
11:             end if
12:         end for
13:         add $a_c$ to $n_1.alert$
14:     end for
15: end if
16: return $S$
```

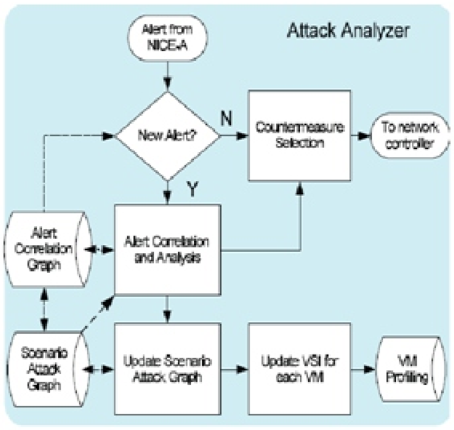
3. 5. Attack Analyser:

The better functions of Precise system are performed by assail analyser, which includes procedures such as flack graph artifact and update, alarum reciprocality, and countermeasure selection. The cognition of constructing and utilizing the SAG consists of figure phases: Aggregation gathering, knock illustration mentation, and possible use itinerary analysis. With this collection, move paths can be modelled using SAG. Apiece client in the onset illustration represents the utilize by the wrongdoer. Apiece track from an initial computer to a goal thickening represents an undefeated beginning. In summary, City act

•Cloud system information is collected from the node controller includes the number of VMs in the cloud server, running services on each VM, and VM's Virtual Interface (VIF) information.

•Virtual network topology and configuration information is collected from the network controller, which includes virtual network topology, host connectivity, VM connectivity, every VM's IP address, MAC address, port information, and traffic flow information.

•Vulnerability information is generated by both on demand vulnerability scanning (i.e., initiated by the network controller and NICE-A) and regular penetration testing using the well-known vulnerability databases, such as Open Source Vulnerability Database (OSVDB) , Common Vulnerabilities and Exposures List (CVE), and NIST National Vulnerability Database (NVD).



3.5 Network controller :

The mesh mechanism is a key part to reason the programmable networking capability to see the realistic scheme reconfiguration lineament supported on Open-Flow prescript. In Pleasant, within each cloud computer there is a software control, for warning, OVS, which is utilized as the provide switching for VMs to handle traffic in and out from VMs. The connection between darken servers (i.e., bodily servers) is handled by physiological Unobstructed Flow-capable Shift (OFS).

3.6. Counter Measure Selections :

Algorithm 2 presents how to superior the best countermeasure for a precondition operation scenario. Input to the formula is a preparedness, attempt illustration G, and a wager of furniture measures CM. The formula starts by selecting the knob v-Alert that corresponds to the alarum generated by an ICE-A. Before selecting the countermeasure, we approximate the distance of v-Alert to the target convexity.

```
Algorithm 2. Countermeasure_Selection
Require: $Alert, G(E,V), CM$
 1: Let $v_{Alert}$ = Source node of the $Alert$
 2: if Distance_to_Target($v_{Alert}$) > $threshold$ then
 3:     Update_ACG
 4:     return
 5: end if
 6: Let $T = Descendant(v_{Alert}) \cup v_{Alert}$
 7: Set $Pr(v_{Alert}) = 1$
 8: Calculate_Risk_Prob($T$)
 9: Let $benefit[|T|,|CM|] = \emptyset$
10: for each $t \in T$ do
11:     for each $cm \in CM$ do
12:         if $cm.condition(t)$ then
13:             $Pr(t) = Pr(t) * (1 - cm.effectiveness)$
14:             Calculate_Risk_Prob($Descendant(t)$)
15:             $benefit[t, cm] = \Delta Pr(target\_node)$.
16:         end if
17:     end for
18: end for
19: Let $ROI[|T|,|CM|] = \emptyset$
20: for each $t \in T$ do
21:     for each $cm \in CM$ do
22:         $ROI[t, cm] = \frac{benefit[t,cm]}{cost.cm + intrusiveness.cm}$.
23:     end for
24: end for
25: Update_SAG and Update_ACG
26: return Select_Optimal_CM($ROI$)
```

If the length is greater than a threshold reckon, we do not full-fill countermeasure option but update the ACG to record itinerary of alerts in the group (parentage 3). For the maker computer v-Alert, all the reachable nodes (including the germ node) are composed into a set T (piping 6). Because the preparedness is generated exclusive after the attacker has performed the production, we set the chance of v-Alert to 1 and designate the new probabilities for all of its kid (downstream) nodes in the set T (lines 7 and 8).

Now, for all t 2 T the applicatory countermeasures in CM are chosen and new probabilities are deliberate according to the strength of the designated countermeasures (lines 13 and 14). The modify in amount of point knob gives the aid for the applied countermeasure using (7). In the incoming equivocal for-loop, we compute the ideal of ROI, is regarded as the best countermeasure. Finally, SAG and ACG are also updated before terminating the algorithm.

4.CONCLUSION :

In this paper, we presented Pleasant, which is planned to sight and mitigate collaborative attacks in the darken realistic networking surroundings. Prissy utilizes the onrush represent worthy to lead onslaught detection and forecasting. The proposed solution investigates how to use the programmability of software switches-based solutions to modify the discovery truth and veto victim victimisation phases of collaborative attacks. The scheme action judgment demonstrates the practicability of Metropolis and shows that the planned set can significantly decrease the peril of the darken scheme from being victimized and abused by intimate and extrinsic attackers.

RERENCES:

1. Juan Du, Daniel J. Dean, Yongmin Tan, XiaohuiGu, and Ting Yu, "Scalable Distributed Service Integrity Attestation for Software-as-a-Service Clouds" in IEEE Transactions On Parallel And Distributed Systems, Vol. 25, No. 3, March 2014.

2. J.Sasidevi, Dr. R. Sugumar, P. ShanmugaPriya "New-Multi-Phase Distribution Network Intrusion Detection" in International Journal of Advanced Research in Computer Science and Software Engineering Volume 5, Issue 3, March 2015

3. J. Du, X. Gu, and T. Yu, "On Verifying Stateful Dataflow Processing Services in Large-Scale Cloud Systems," Proc. ACM Conf. Computer and Communications Security (CCS), pp. 672-674, 2010.

4. I. Hwang, "A Survey of Fault Detection, Isolation, and Reconfiguration Methods," IEEE Trans. Control System Technology, vol. 18, no. 3, pp. 636-653, May 2010.

5. 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.

AUTHORS:

M. Lakshmi Priyanka, I have received my B.Tech Degree in Computer Science & Engineering stream from the Siddharth InstituteofEngineering& Technology, Puttur Affiliated to the Jawaharlal Nehru technological university,Ananthapur, A.P, India in 2012, and pursuing M.Tech in Computer Science Engineering from the Audhisankara College of Enginerring and Technology (Autonomous), Gudur, Affiliated to the Jawaharlal Nehru technological university, Ananthapur now.

B.Sudhakar, He has received his B.Tech in Computer science and Engineering from Sree Venkateswara University, Tirupathi, and M.Tech (PG) in Computer Science and Engineering from JNTU College of Hyderabad, Hyderabad, A.P, India . He is dedicated to teaching field from the last 12 years. He has guided 6 P.G Students and 10 batches U.G students. At present he is working as Assistant College of Enginnering and Technology, Gudur, S.P.S.R Nellore(Dt), Andhra Pradesh, India.