2013 IEEE International Conference on Cloud Computing Technology and Science

Community Cloud: Concept, Model, Attacks and Solution

Hasen Nicanfar, Qiang Liu, Peyman TalebiFard, Wei Cai, Victor C.M. Leung
hasennic@ece.ubc.ca, libra6032009@gmail.com, peymant@ece.ubc.ca, weicai@ece.ubc.ca, vleung@ece.ubc.ca

***Abstract*—In this paper, we propose a new model of community cloud (ComC) interaction that is based on the demand of tenants and propose an intrusion detection mechanism for the proposed model. ComC is a solution that is more secure than the public cloud, and less costly than the private cloud. We argue that our proposed model of the ComC will be more beneficial to consumers as well as providers. Our evaluation shows the efficiency of the proposed model from cost and operation point of views. In addition, our analysis shows that our proposed IDS can make the ComC a safe environment and can guarantee the security and privacy of the customers.**

***Keywords*-Community Cloud; Security; Privacy; IDS.**

I. Introduction

The main idea behind the Cloud Computing (CC) came from having the computing as a utility. Organizations and consumers can buy most of the services that on a need basis from the service providers at a reasonable cost. CC can reduce the capital expenditure of new businesses and offer a more cost efficient solution for large enterprises. It is important to have a model that organizations can benefit from the benefits of public cloud in terms of billing and pay-as-you-go with added level of privacy, security and policy management. Community cloud (ComC) is a collaborative solution towards a multi-tenant infrastructure shared among different organizations that can be managed internally or by a third party organization. Privacy and security of the cloud based services and applications has gained attention of the CC stakeholders including users, customers, providers, society and governments, as well as the research community.

Although the CC market is growing fast and most of the computing services are being provided and purchased in this market, the development of the required privacy and security systems are far behind. Maybe this is the main reason that we only have a few main service providers in the CC market, precisely, at a public cloud e.g. Amazon, Microsoft and Google. On the other hand, lack of a robust and trusted privacy and security system motivated the idea of private cloud and hybrid cloud. One of the non-public or semi-public/-private model of the cloud, which is our concentration in this paper, is ComC. Precisely, ComC is introduced to provide cloud based services to limited customers with similar privacy and security concerns [1].

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada, TELUS, Institute for Computing, Information and Cognitive Systems, and National Natural Science Foundation of China, under grant Nos. 61170287, 60970034 and 61070198.

CC still suffers from various privacy concerns and attacks such as IP spoofing, Address Resolution Protocol spoofing, Flooding, Denial of Service (DoS), Distributed DoS, etc. In order to tackle and mitigate them, efficient intrusion detection systems (IDS) and intrusion prevention systems should be incorporated in CC infrastructures. The IDS is a software or hardware that automatically detects potential threats toward a target computer system or network via analyzing multi-source data, e.g. network traffic, audit data, system logs. Typically, an IDS consists three main modules, such as monitoring, analyzing and decision, which are respectively responsible for collecting raw data, preprocessing or formatting multi-source data, and determining if a threat has occurred. The detection methodologies of the IDS are generally categorized into three groups, namely signature-, anomaly- and specification-based detections, each one has its own advantages and drawbacks in terms of the detection capacity, false alarms, overheads and scalability [2].

Contribution: We make a proposition on the model of ComC and discuss the efficiency of the model in terms of cost and operation from consumer and provider perspectives. We analyze attacks against the model following by proposing a service-based IDS (SbIDS) to protect the model.

In the existing model of ComC, homogeneity of the customers that are using the same ComC is investigated. However, we argue that heterogeneity of the customers can also be a key element in the success of ComC. This will make ComC more efficient from business and technical point of views. Based on current state-of-the-art, we study the attacks that are targeting CC and are the subject of ComC. In addition, due to the nature of having multi-customer model of a private cloud as ComC, the attacks that can be performed specifically on ComC through other ComC customers are also being considered. Finally, to protect ComC against the attacks, we proposed a service based IDS.

Section II describes literature review and our ComC model and our SbIDS are presented in Section III and analyzed in Section IV. The paper is concluded in Section V.

II. Literature Review

Although there are many definitions for the cloud that may presented from different point of views, they all follow the same concept of "*Computing as a Utility*'. In some, attention is paid to the computing/software, networking, data warehousing (Data Center), or combination of them. Our main references for the definitions are the US National

978-0-7695-5095-4/13 $31.00 © 2013 IEEE
DOI 10.1109/CloudCom.2013.163

Table I: Summary of the comparison

Feature	Public cloud	Private cloud	Hybrid cloud	ComC-C	ComC-S
Number of Customers	Unlimited	One	Unlimited	Limited	Limited
Cost (price of service)	Low	High	Medium	Medium	Medium
Fast scalability	Very high	Low	High	Medium	Medium
Resource Utilization	Very high	Low	High	Medium	Medium-High
Security, Privacy & Trust	Low	High	Medium-High	Medium	Medium-High

Institute of Standards and Technology (NIST), the National European Network and Information Security Agency (ENISA) and Cloud Security Alliance (CSA) [1], [3], [4]. As per NIST, CC is "*a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (for example, networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction*". ENISA defines CC as "*an on-demand service model for IT provision, often based on virtualization and distributed computing technologies*".

In [5], Internet-of-Services (IoS) is introduced for CC that covers three main identified services of the cloud, such as SaaS (Software-as-a-Service), PaaS (Platform-as-a-Service) and IaaS (Infrastructure-as-a-Service) [1]. Indeed the IoS concepts introduce any Information and Communication Technology (ICT) that can be delivered by CC, e.g. security-as-a-service, privacy-as-a-service, and trust-as-a-service. Moreover, five main roles have been identified such as cloud customer, cloud provider, cloud broker, cloud auditor and cloud carrier [6], where the detail duty of of each role can be found in [1]. Since security, privacy and trust are of the main concerns by most of the customers, private cloud solutions tend to become more attractive. In fact, a provider in the private cloud model delivers the required services only to one customer. Therefore, the customer can force the appropriate privacy and security mechanisms to improve the level of safety and protection. However, since there is only one customer, cost of service is high, and provider is less flexible on expanding the required service on-the-fly, comparing to the public cloud. In a hybrid model, the customer uses private cloud for high sensitive task and data, and uses public cloud for the low sensitive task and data. TABLE I presents a summery of this analysis.

In ComC, number of customers are limited, such as research departments, and indeed it is defined for the customers with similar concerns, e.g. security and privacy. The definition of ComC mentioned by NIST [1] that is also used in [7], is called ComC-S in our summery presented in TABLE I. However, there is another model in the literature, in which customers give their extra resources to the community, and receive the service when they need [8]. Since the service is actually provided by a customer to other customers, we show this model ComC-C in TABLE I. ComC has recently been emerging as a considerable solution to share the underutilized resources among multiple clouds with distinct security levels and reliability requirements while to ensure an acceptable level of the overall security and privacy [8]. However, the threats in the conventional clouds challenge ComC as well. Apart from the data security [9], the system and behaviour security are also vital for secure ComC. The IDS for clouds [10], especially for ComC [7], [11], can serve as the second line of defence to protect clouds from diverse internal and external threats. The authors in [10] claimed that extensibility, compatibility and efficient management to virtualization-based context needed to be introduced into existing IDS implementations. They summarized requirements for deploying IDS in the cloud.

Regarding the security mechanisms corresponding to ComC, the authors in [7] proposed a software architecture named Virtual Interacting Network Community (Vinci) that exploited virtualization to secure ComC. In the architecture, a community defined several overlays by instantiating and interconnecting virtual machines (VMs) that were defined from a small set of templates. Vinci included templates to run user applications, protected shared resources and controlled traffic among communities.Furthermore, in [11] the resilience of IT services are studied when regional catastrophic events occurred and proposed utilizing ComC to improve resilience of businesses after the events. Indeed, the communities geographic diversity in regions gave businesses a chance of re-establishing operations after a catastrophic event, meeting the objective of business resilience.

Our study shows that privacy is one of the main issues in the CC [12]. Privacy and security in the (public) cloud is studied in [13] as the main concerns. Moreover and in [14], authors focused on access control in the cloud with privacy preserving concern. in [10], an IDS in the cloud is proposed. [15] discusses the security and privacy in the cloud and [16] focused on trust problem and modelling it in the cloud. The concentration of [5], [6] is discussing the open issues in the cloud about the security and privacy. Although our studies as partially discussed above, shows that security and privacy in the cloud has gained attention of the research community during last few years, market is ahead of the research community and the cloud is being implemented and being used with all of the open privacy and security issues. There are many venues, e.g. conferences and journal, that are initiated helping the research in this area; however, more works need to be done to reach to an accepted and reasonable level of security and privacy in CC.

III. Our designed system

A. Community cloud model

Let us consider the cloud presented in Figure 1, with a limited customers, e.g. N customers (Cst_n , $n = 1, ..., N$). Also, let us assume the number of provided services are M services (Srv_m , $m = 1, ..., M$).note that, in this figure, we have another layer for the SbIDS agents, which will be explained in the next sub-section.

Definition: Let us define function $DmdC(Cst_n)$, as per (1), that shows the amount ($\alpha^n_{(m,u)}$) of unit service Srv_m per time unit u required/requested by the customer Cst_n.

$$\begin{cases} DmdC(Cst_n) = (\alpha^n_{(m,u)}, Srv_m) \\ where \quad m = 1, ..., M \ \& \ u = 1, ..., U \end{cases} \tag{1}$$

In fact, the demand function $DmdC(Cst_n)$ is a matrix $M \times U$ per each customer Cst_n, as follow:

$$DmdC(Cst_n) = \begin{bmatrix} \alpha^n_{(1,1)} & \cdots & \alpha^n_{(1,U)} \\ \vdots & \ddots & \vdots \\ \alpha^n_{(M,1)} & \cdots & \alpha^n_{(M,U)} \end{bmatrix}$$

In this matrix, each row represents a service Srv_m and each column represents a time unit ($u = 1, ..., U$).

Definition: Let us define function $DmdS_u(Srv_m)$ as per (2), which shows the total unit amounts of service Srv_m per time unit u that is required/requested by the entire customers. Also, $DmdS(Srv_m)$ via (3) or (4), is the total required amounts of service Srv_m by the entire customers for during the total time units.

$$DmdS_u(Srv_m) = \sum_{n=1}^{N} \alpha^n_{(m,u)} \tag{2}$$

$$DmdS(Srv_m) = \sum_{u=1}^{U} DmdS_u(Srv_m) \tag{3}$$

$$DmdS(Srv_m) = \sum_{u=1}^{U} \sum_{n=1}^{N} \alpha^n_{(m,u)} \tag{4}$$

In order for the service provider of ComC to meet the requested demands all the time, as per (2), the total demand of a service Srv_m per time unit u should be considered as the lower bound of the proposed service by the service provider, as it is shown by $DmdS_{MAX}(Srv_m)$ in (5).

$$\begin{cases} DmdS_{MAX}(Srv_m) = max \quad DmdS_u(Srv_m) \\ \quad where \quad u = 1, ..., U \end{cases} \tag{5}$$

In fact, the service provider should have the appropriate resource that can deliver the demand for maximum requested of the service. Note that in an optimum situation, the total prepared service Srv_m can be equal to the maximum of the demands in a time unit for that service.

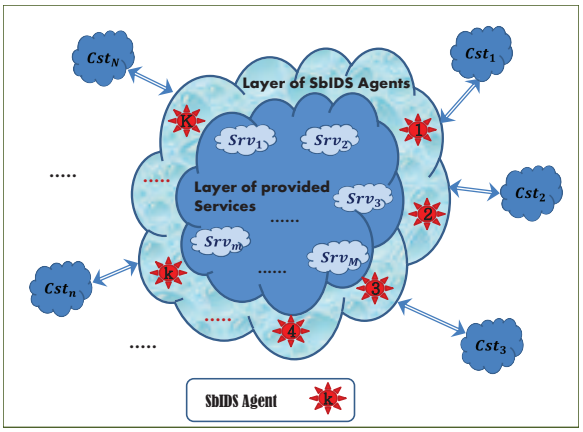


Figure 1: Overall model of the ComC

Definition: Let us present the resource/service utilization by $Utlz(Srv_m)$, which can be calculated through (6).

$$Utlz(Srv_m) = \frac{DmdS(Srv_m)}{U \times DmdS_{MAX}(Srv_m)} \tag{6}$$

Note: The best situation for the service provider is maximizing the resource (service Srv_m) utilization. which consequently enables the service provider to deliver the service at a lower rate/price. Hence, as (6) shows, the maximizing the utilization of each resource/service can be pronounced by minimizing the $DmdS_{MAX}(Srv_m)$ and maximizing the $DmdS(Srv_m)$. There is an upper bound per each time unit for each resource without adding the new resource. Therefore, to maximize the utilization in total, the business model should be able to maximize the resource/service utilization per time unit (any small time unit). As a result, the best model of accepting the customers is having a basket of the customers with different demands per time unit. Although the customers' concern about the privacy and security is the main criteria to design and build the ComC for, choosing customers with different business model and mix demands is the key factor for the better resource utilization yields to a cost efficient as well as better/lower service price.

Remark: Hence and to conclude above analysis and discussion, the best and efficient model for a ComC is having a mix of customers with non-equal demands per service at any given time (unit). We will analyze this in Section IV.

B. SbIDS

ComC is under attack by most of the well-known attacks of the cloud, since ComC is a shared resource environment. Therefore, attacks e.g. VM neighbour or other CC attacks are applicable. One of the ComC specific attacks is DoS attack although a DoS attack may not affect the public cloud. In the public cloud, if the adversary attacks the system by targeting any service e.g. by a DoS attack, the service provider can increase the service availability shortly to keep the other

customer satisfaction, and then catch the adversary and fix the issue. However, due to the limited resources in ComC and private cloud, DoS is more effective on the private cloud and then ComC. Since the main element in CC is a service, so, an adversary can attack the system, ComC, based on the service. Therefore, we follow the same concept and propose our IDS as a service-based. In fact, our IDS, precisely SbIDS, considers attacking the provided services in ComC. Our literature review shows that main referred IDSs similarly consider service as the main factor in CC.

As per hown topology in Figure 1, we take advantage of auditor role of the cloud [1], [6] in order to design a ring for monitoring/controlling accessing to the services in ComC. This design consists of a series of agents Agn_k , $k = 1, ..., K$ for ComC. Each agent monitors accessing to a service by each customer and sends the auditing reports to the main SbIDS control engine, as part of auditor role of the cloud. Since a customer may receive and have access to verity of the services in ComC, e.g. IaaP and SaaS, each agent also monitor the appropriate service. To make a general format and align with our above discussion, we assume an agent is only monitoring one service, e.g. Srv_n. Indeed, we may have multiple agents to monitor accessing to the same service Srv_n, based on the ComC size.

Our framework is a service-based model, and is designed referring to service architecture discussed in [17], and our previous proposals in [18], which is presented in Figure 2. To design our framework, and detail of the algorithm, we refer to the transparency concept in the CC, as per [4], and precisely we follow the CSA protocol in [19]. Our algorithm to setup and proceed the detecting and controlling intruder as per of the SbIDS is presented by Algorithm 1.

As our algorithm shows, first of all, a new (potential) customer gets set up by contacting the service registry and granted permission to have access to the service(s). The initial demands sent by the customer is for starting the deal, which will be saved in the database of the ComC demand. However, the customer may increase the demand later, in which SbIDS will catch the extra request as a suspicious request. The SbIDS controller confirms the extra demand with the customer, and if it is a valid request, the demand will be handled and service registry updates the ComC demand database accordingly. In case of not confirming the demand, it will be considered as an attacks and the communication will be stopped till the issue gets fixed. In case of valid request, service provider may need to increase overall resources to adjust extra service request, which is obvious that it can change the base of the service price.

Note that in above discussion, we only explained extra service request by a customer, which indeed can be part of defending the ComC against DoS. However, an agent can observe any misbehaving of a customer and reports it to the controller. In fact, the system agents use the signature database to figure out if the customer is attacking the system.

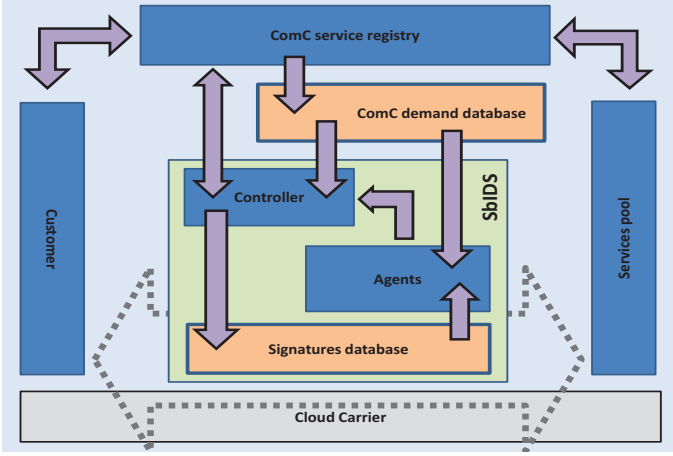


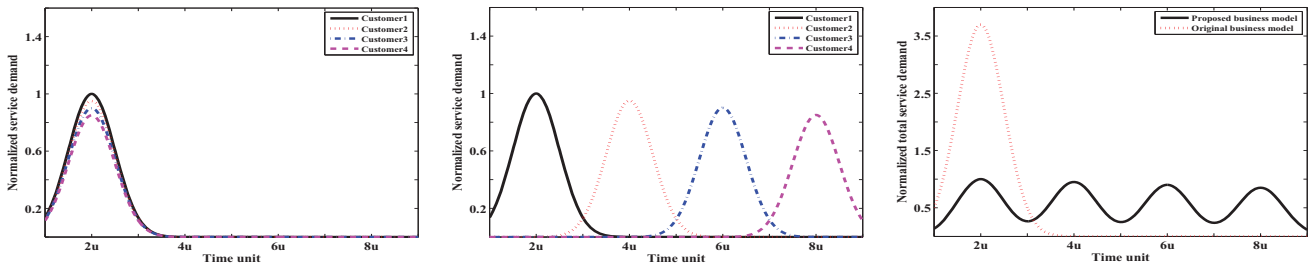
Figure 2: Service-based intrusion detection system

Algorithm 1 SbIDS

Define:
$ComCReg.$: Community cloud service registry.
Prv : Cloud service provider.
Cst_n : Cloud customer, $n = 1, ..., N$
Srv_m : Cloud service, $m = 1, ..., M$.
Agn_k : SbIDS agent, $k = 1, ..., K$.
$\alpha^n_{(m,u)}$: Requested unit service Srv_m per time unit u by customer Cst_n.
$DmdC(Cst_n)$: Detail demands of customer Cst_n.
$DmdS_{MAX}(Srv_m)$: Maximum demand per time unit of Srv_m.
$SLA(DmdC(Cst_n))$: Service Level Agreement for customer Cst_n that meets detail demand $DmdC(Cst_n)$.

Setup:
Cst_p (new potential customer) sends request-for-information (RFI) to $ComCReg.$ of the ComC, for current information.
$ComCReg.$ replies back to Cst_p by list of customer (Cst_n & $n = 1, ..., N$) and list of proposed services (Srv_m & $m = 1, ..., M$).
Cst_p investigates the received information, and ends if the ComC is not safe, otherwise goes to next step.
Cst_p sends request-for-proposal/quote (RFP/RFQ) along with $DmdC(Cst_p)$ to $ComCReg.$.
$ComCReg.$ informs current customer (Cst_n) as well as Prv(s) about the potential customer Cst_p and list of requested services by the new customer.
Current customers and provider(s) inform $ComCReg.$ if any issue.
In case of having conflict-of-interest or any issue raised by current customers and/or Prv, $ComCReg.$ informs Cst_P and declines the request, and ends the deal. Otherwise, goes to next step.
$ComCReg.$ checks each row of the demand matrix of the new customer, and compares it with the current load per each service (Srv_m).
$ComCReg.$ sets the price with respect to amount of increasing the $DmdS_{MAX}(Srv_m)$, if any. Accordingly, $ComCReg.$ prepares an $SLA(DmdC(Cst_p))$ and sends it to the potential customer Cst_p.
If Cst_p agrees on the received SLA ($SLA(DmdC(Cst_p))$), confirms it with the $ComCReg.$, in which ComC $ComCReg.$ adds Cst_p to the list of valid customers as $Cst_{N+1} = Cst_p$, in the ComC demand database.

Monitoring/Controlling
SbIDS agents monitor accessing the services by the customers.
The agent checks the service access/request versus demand of the customer saved in customer profile in the ComC demand database.
The agent also checks the service access/request versus intrusion signature(s) in the Signature database of the SbIDS.
The agent reports to the SbIDS controller if any conflict, from above two steps of checking/monitoring steps.
In case of an issue, SbIDS informs customer the changes and adjusts the ComC demand database (customer profile) if the change is not an attack. In this case, communication is via ComC service registry, in which the price may need to be adjusted due to increasing the provided resource/service, if any.
If the extra demand is part of an attack, SbIDS cancels/rejects the request and goes to protection mode to fix the issue.
Controller updates the signature database in case of finding a new attack.



(a) based on the original business model (b) based on the propose business model (c) the comparative results of the total value

Figure 3: The normalized service demand of four customers versus the time unit

IV. Evaluation and Analysis

A. Performance Evaluation

In order to perform a fairness analysis, we define two scenarios, one based on original business model (Scn1) and second one based on our proposed model (Scn2). For simplicity, we assume four customers that have service demands with the same standard deviation σ but with the different normalized value of the maximum demand. The normalized service demand of a customer is defined by:

$$DmdS_i(t) = MaxDmd_i \times e^{\frac{-(t-t_0)^2}{2\sigma^2}} \quad (7)$$

where $MaxDmd_i$ denotes the normalized value of the maximum demand of the customer i. Considering the worst case where all customers reach their maximum demand at the same time, denoted by t_0, we compare the proposed business model with the original one to demonstrate the advantages of the proposed model in terms of the service fairness. Here, we set the value of t_0 to $2u$ and select the maximum service demands of these customers as $MaxDmd_1 = 1$, $MaxDmd_2 = 0.95$, $MaxDmd_3 = 0.9$ and $MaxDmd_4 = 0.85$.

Figure 3a and 3b respectively illustrate the normalized service demand of four customers versus the time unit based on the original and the proposed business models. We see in Figure 3a that all customers in the worst case reach their maximum service demand at the time of $2u$, as described above. However, Figure 3b shows that the maximum service demands of these customers are adjusted to different time units. The reason is that the original business model provides services for customers in an on-demand manner without considering the fairness of resource utilization, whereas the proposed business model take the fairness into consideration by scheduling services at different time units in order to sufficiently use the un-utilized system resources. Therefore, the total value of the normalized service demands for all customers based on the proposed business model is much smaller than that based on the original model, as illustrated in Figure 3c. The comparative results highlight the fact that the proposed business model achieve better fairness compared to the original model, reducing the resource requirement for serving all customers.

B. Security analysis

We present two adversary models consists of internal and external where we consider Dolev-Yao proposal [20]. To adapt it to the service-based environment, we assume the message is access to a service. All of the requests or service access messages generated by the honest customers and any other parties are sent to the adversary, and the honest entities receive the messages only from the adversary.

1) Internal Adversary Model: In this model, our adversary is one of the ComC customers.

Objective: Objective of the adversary can consists of: (i) gaining access to other customers' data, (ii) interfering to other customers' accessing to a service, (iii) overloading the service to make it inaccessible at the time requested by others, e.g. by performing a DoS attack.

Initial capability: The adversary knows the detail design of our system and framework. He also knows other active ComC customers, their requested services, as well as list of available and provided services in ComC. Note that he does not know the detail demand of others.

Capability during the attack: Similarly, he can increase or decrease his demand per time unit, or may ask a new service that was not originally in his matrix of demand, and approve the changes.

Discussion: One of the key information required by the adversary on attacking a victim is the detail demand of the victim. Precisely, although he knows the requested services by the victim, he does not know the detail demand per time unit of the victim. He may perform DoS in different time unit and try to observe the detail demand of the victim. However, firstly this attack is costly since at each changing and requesting extra service, he should approve them as per our algorithm, and pay the cost. Secondly, he may find the maximum demand in a specific time unit, but he cannot observe which customer is demanding. Finally, since he should perform the attack in different time and sweep the entire time units, his behaviour can be noticed by the SbIDS agent, and controller, while the controller can add a behaviour signature to the signature database to stop him.

2) External Adversary Model: Our adversary is not any of the system entities, therefore he has less information and has limited knowledge comparing to the internal adversary.

Objective: The adversary wants to gain access to one of the ComC customers, data or task/job. He may also wants to perform a DoS attack to overload the system.

Initial capability: Similarly, the detail information about our system design model in known by him.

Capability during the attack: He can send any request consists of any kind of demand matrix to ComC.

Discussion: First of all, unless the adversary becomes one of the trusted customer, voted by the current ComC customers, he can not gain access to any service, and then he will be considered as an internal adversary. To obtain information about the current customers of the ComC, he may send a dummy request, and not finalized the deal at the end. In this case and as per SLA, he will have the required information. However, his ability to use the received information about ComC is less than internal adversary. If he performs a DoS attack and sends several requests, his misbehaving can be noticed by the agent and controller of the SbIDS and referred it to a mis-behaviour signature.

V. Conclusion

We proposed a new model for ComC that considers heterogeneity of tenants while taking the demand patterns into account. We have then proposed an IDS for the proposed model, which is based on a SOA approach. This framework has influenced the service-based IDS at the framework level. We proposed the algorithm of joining and set up of a new customer as well as controlling and managing access to the ComC services and protecting them from the attacks. Our analysis showed that our ComC model is efficient by increasing the service utilization and decreasing the upfront capital expenditure required by the providers. Furthermore, it showed that SbIDS is secure and capable of detecting and stopping attacks from internal and external intruders. To extend this work, we will emphasis on detailed design of the ComC model as well as SbIDS.

References

[1] National Institute of Standard and Technology, "NIST," Website. [Online]. Available: www.nist.gov/

[2] A. PATEL, M. TAGHAVI, K. BAKHTIYARI, and J. CELESTINO, "An intrusion detection and prevention system in cloud computing: A systematic review," *Journal of network and computer applications*, vol. 36, no. 1, pp. 25–41, 2013.

[3] European Network and Information Security Agency, "ENISA," Website. [Online]. Available: www.nist.gov/

[4] Cloud Security Alliance, "CSA," Website. [Online]. Available: www.cloudsecurityalliance.org/

[5] R. Moreno-Vozmediano, R. Montero, and I. Llorente, "Key challenges in cloud computing to enable the future internet of services," 2012.

[6] P. Mell, "What's special about cloud security?" *IT Professional*, vol. 14, no. 4, pp. 6–8, 2012.

[7] F. Baiardi and D. Sgandurra, "Securing a community cloud," in *Distributed Computing Systems Workshops (ICDCSW), 2010 IEEE 30th International Conference on*. IEEE, 2010, pp. 32–41.

[8] B. Saovapakhiran and M. Devetsikiotis, "Enhancing Computing Power by Exploiting Underutilized Resources in the Community Cloud," in *Communications (ICC), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–6.

[9] M. A. AlZain, E. Pardede, B. Soh, and J. A. Thom, "Cloud computing security: from single to multi-clouds," in *System Science (HICSS), 2012 45th Hawaii International Conference on*. IEEE, 2012, pp. 5490–5499.

[10] S. Roschke, F. Cheng, and C. Meinel, "Intrusion detection in the cloud," in *Dependable, Autonomic and Secure Computing, 2009. DASC'09. Eighth IEEE International Conference on*. IEEE, 2009, pp. 729–734.

[11] G. Garlick, "Improving resilience with community cloud computing," in *Availability, Reliability and Security (ARES), 2011 Sixth International Conference on*. IEEE, 2011, pp. 650–655.

[12] N. Kshetri and S. Murugesan, "Cloud computing and eu data privacy regulations," *Computer*, vol. 46, no. 3, pp. 0086–89, 2013.

[13] C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 362–375, 2013.

[14] M. Nabeel and E. Bertino, "Privacy preserving delegated access control in public clouds," *IEEE Transaction on Knowledge and Data Engineering*, 2013.

[15] Z. Xiao and Y. Xiao, "Security and privacy in cloud computing," *accepted for publication in IEEE Communication Surveys & Tutorials*, vol. 15, no. 12, pp. 843–859, Second Quarter 2013.

[16] D. Wallom, M. Turilli, G. Taylor, N. Hargreaves, A. Martin, A. Raun, and A. McMoran, "mytrustedcloud: Trusted cloud infrastructure for security-critical computation and data managment," in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*. IEEE, 2011, pp. 247–254.

[17] T. Erl, *SOA: Principles of service design*. Prentice Hall Press Upper Saddle River, NJ, USA, 2007.

[18] P. TalebiFard and V. C. Leung, "Context-aware mobility management in heterogeneous network environments," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 2, no. 2, pp. 19–32, 2011.

[19] Ron Knode and Doug Egan, "Digital Trust In The Cloud: A Precis for the CloudTrust Protocol (V2.0)," Guidline, July 2010. [Online]. Available: https://cloudsecurityalliance.org/download/a-precis-for-the-cloudtrust-protocol-v2-0/

[20] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.