Detecting Occasional Reputation Attacks on Cloud Services

Talal H. Noor, Quan Z. Sheng, and Abdullah Alfazi

School of Computer Science
The University of Adelaide, Adelaide SA 5005, Australia
{talal,qsheng,abdullah}@cs.adelaide.edu.au

Abstract. Cloud service consumers' feedback is a good source to assess the trustworthiness of cloud services. However, it is not unusual that a trust management system experiences malicious behaviors from its users. Although several techniques have been proposed to address trust management in cloud environments, the issue of how to detect occasional reputation attacks on cloud services is still largely overlooked. In this paper, we introduce an occasional attacks detection model that recognizes misleading trust feedbacks from occasional collusion and Sybil attacks and adjusts trust results for cloud services that have been affected by these malicious behaviors. We have collected a large collection of consumer's trust feedbacks given on real-world cloud services (over ten thousand records) to evaluate and demonstrate the applicability of our approach and show the capability of detecting such malicious behaviors.

Keywords: Trust Management, Cloud Computing, Occasional Attacks, Attacks Detection.

1 Introduction

The highly dynamic, distributed, and non-transparent nature of cloud services makes trust management in cloud environments a challenging problem [10,6,8]. Several techniques have been proposed to assess and manage trust based on feedback collected from participants [6,5,1]. However, not much attention has been given to detect *occasional* and *periodic* reputation attacks on cloud services. The main goal of our work is to detect occasional and periodic reputation attacks on cloud services[1]. Unfortunately, this is not an easy task due to some unique characteristics of cloud environments: i) consumers are dynamic and may have multiple accounts for a particular service (e.g., owning multiple email accounts in Gmail) which makes it difficult for a Trust Management Service (TMS) to detect whether a Sybil attack is performed; ii) the occasional way that these attacks occur, as described in [10], which makes the detection of such malicious behaviors a significant challenge and significantly affects the performance of TMS. TMS

[1] Other techniques for detecting attacks on different distributions over an extended period of time are previously proposed. Interested readers are referred to [9,8] for more technical details.

F. Daniel, P. Dolog, and Q. Li (Eds.): ICWE 2013, LNCS 7977, pp. 416–423, 2013.

should be able to efficiently detect such attacks and thus dilute the influence of those misleading feedbacks to enable more robust trust calculations.

In this paper, we overview the design and implementation of the occasional attacks detection model. This model allows TMS to detect misleading feedbacks from collusion and Sybil attacks and helps to have robust trust calculations. In a nutshell, the salient features of the model are i) *Occasional Collusion Attacks Detection Metric*: this metric distinguishes between misleading and credible feedbacks by detecting the occasional collusion attacks (i.e., attackers who intend to manipulate the trust results by giving multiple trust feedbacks to a certain cloud service in a short period of time [3]); ii) *Occasional Sybil Attacks Detection Metric*: this metric allows TMS to identify misleading trust feedbacks from Sybil attacks and detect occasional Sybil attacks (i.e., attackers who create multiple identities and leave misleading trust feedbacks in a short period of time to trick cloud service consumers into trusting cloud services that are not trustworthy [4]); iii) *Adaptivity and Flexibility*: the model is adaptive and flexible in the sense that it is possible to tweak the metrics according to the trust evaluation needs (e.g., to detect the collusion attacks only or to detect both attacks).

The remainder of the paper is organized as follows. Section 2 details the trust management service including the trust feedback collection and assessment and briefly describes the identity management service. Section 3 describes the details of our occasional attacks detection model. Section 4 reports the implementation and several experimental evaluations. Finally, Section 5 discusses the related work and provides some concluding remarks.

2 Trust Management Service (TMS)

In a typical reputation-based TMS, consumers either give feedback regarding the trustworthiness of a particular cloud service or request trust assessment for the service[2]. From consumers' feedback, the trust behavior of a cloud service is represented by a tuple $\mathcal{H} = (\mathcal{C}, \mathcal{S}, \mathcal{F}, \mathcal{T}_f)$, where $\mathcal{C}$ is the consumer's primary identity, $\mathcal{S}$ is the cloud service's identity, and $\mathcal{F}$ is a set of feedbacks (i.e., based on several Quality of Service (QoS) parameters including availability, security, response time, etc.). Each feedback in $\mathcal{F}$ is represented in numerical form with the range of [0, 1], where 0, 1, and 0.5 means *negative*, *positive*, and *neutral* feedback respectively. $\mathcal{T}_f$ is the timestamps when feedbacks are given. TMS calculates the trust result, denoted as $\mathcal{T}_r(s)$, from the collected feedbacks as follows:

$$\mathcal{T}_r(s) = \frac{\sum_{c=1}^{|\mathcal{V}(s)|} \mathcal{F}(c,s) * \mathcal{O}_a(s,t_0,t)}{|\mathcal{V}(s)|} + \chi * \mathcal{C}_t(s,t_0,t) \tag{1}$$

where $\mathcal{V}(s)$ denotes feedbacks given to the cloud service s and $|\mathcal{V}(s)|$ represents the total number of trust feedbacks. $\mathcal{F}(c,s)$ are feedbacks from the c^{th} consumer weighted by the occasional attacks detection factors $\mathcal{O}_a(s,t_0,t)$ to allow TMS to dilute the influence of misleading feedbacks. $\mathcal{F}(c,s)$ is held in the invocation

[2] We assume a transaction-based feedback where all feedbacks are held in the TMS.

history record h and updated in TMS. $\mathcal{C}_t(s, t_0, t)$ is the change rate of trust results in a period of time that allows TMS to adjust trust results for cloud services that have been affected by malicious behaviors. χ is the normalized weight factor for the change rate of trust results which increase the adaptivity where the higher χ is, the more the cloud service is rewarded and *vice versa*. More details on how to calculate $\mathcal{O}_a(s, t_0, t)$ and $\mathcal{C}_t(s, t_0, t)$ are described in Section 3.

Since trust and identification are closely related [2], the Identity Management Service (IdM) can facilitate TMS in the detection of occasional Sybil attacks against cloud services without breaching the privacy of consumers. When consumers attempt to use TMS for the first time, they are required to register their credentials at the trust identity registry in IdM to establish their identities. The trust identity registry stores an identity record represented by a tuple $\mathcal{I} = (\mathcal{C}, \mathcal{C}_a, \mathcal{T}_i)$ for each consumer. $\mathcal{C}$ is the consumer's primary identity. $\mathcal{C}_a$ represents a set of credentials' attributes (e.g., passwords, IP address, etc.) and $\mathcal{T}_i$ represents the consumer's registration time in TMS. More details on the detection of occasional Sybil attacks can be found in Section 3.

3 Occasional Attacks Detection Model

Occasional Collusion Attacks Detection Metric. We consider *time* in detecting occasional and periodic collusion attacks (i.e., periodicity). In other words, we consider the total number of feedbacks $|\mathcal{V}(s)|$ given to cloud service s during a period of time $[t_0, t]$. The sudden change in the feedback behavior indicates an occasional feedback collusion. To detect such behaviors, we measure the percentage of occasional and periodic change in the total number of trust feedbacks among the whole feedback behavior (i.e., consumers' behavior in giving feedbacks for a certain cloud service). The occasional feedback collusion factor $\mathcal{O}_f(s, t_0, t)$ of cloud service s in a period of time $[t_0, t]$, is calculated as follows:

$$\mathcal{O}_f(s,t_0,t) = 1 - \left(\frac{\left(\int_{t_0}^{t} |\mathcal{V}(s,t)| \, \mathrm{d}t \right) - \left(\int_{t_0}^{t} \Delta_f(s,t) \mathrm{d}t \right)}{\int_{t_0}^{t} |\mathcal{V}(s,t)| \, \mathrm{d}t} \right)$$
$$where \Delta_f(s,t) = \begin{cases} \mathcal{C}\mu\left(|\mathcal{V}(s,t)|\right) & if\, |\mathcal{V}(s,t)| \geq \\ & \mathcal{C}\mu\left(|\mathcal{V}(s,t)|\right) \\ |\mathcal{V}(s,t)| & otherwise \end{cases} \tag{2}$$

where the first part of the numerator represents the whole area under the curve which represents the feedback behavior for cloud service s. The second part of the numerator represents the intersection between the area under the curve and the area under the cumulative mean of the total number of feedbacks $\mathcal{C}\mu\left(|\mathcal{V}(s,t)|\right)$ (i.e., which represents the mean of all points in the total number of feedbacks and up to the last element because the mean is dynamic and changes from time to time). The denominator represents the whole area under the curve. As a result, the higher the occasional change in the total number of feedbacks, the more likely that the cloud service has been affected by occasional collusions.

Occasional Sybil Attacks Detection Metric. Malicious users may manipulate trust results to disadvantage particular cloud services by creating multiple accounts and giving misleading feedbacks in a short period of time (i.e., Sybil attacks). To overcome the occasional Sybil attacks, we consider the total number of established identities $|\mathcal{I}(s)|$ for consumers who gave feedbacks to cloud service s during a period of time $[t_0, t]$. The sudden changes in the total number of established identities is an indicator for an occasional Sybil attack. To detect such behavior, we measure the percentage of occasional and periodic change in the total number of established identities among the whole identity behavior (i.e., all established identities for consumers who gave feedbacks to a particular cloud service). The higher the change in the total number of established identities, the more likely that the cloud service has been attacked by an occasional Sybil attack. Similarly, the occasional Sybil attacks factor $\mathcal{O}_i(s, t_0, t)$ of a certain cloud service s in a period of time $[t_0, t]$, is calculated as follows:

$$\mathcal{O}_i(s,t_0,t) = 1 - \left(\frac{\left(\int_{t_0}^{t} |\mathcal{I}(s,t)| \, \mathrm{d}t\right) - \left(\int_{t_0}^{t} \Delta_i(s,t)\mathrm{d}t\right)}{\int_{t_0}^{t} |\mathcal{I}(s,t)| \, \mathrm{d}t} \right)$$
$$where \Delta_i(s,t) = \begin{cases} \mathcal{C}\mu\left(|\mathcal{I}(s,t)|\right) & if\ |\mathcal{I}(s,t)| \geq \\ & \mathcal{C}\mu\left(|\mathcal{I}(s,t)|\right) \\ |\mathcal{I}(s,t)| & otherwise \end{cases} \tag{3}$$

Based on the proposed occasional attacks detection metrics, TMS dilutes the influence of those misleading feedbacks by assigning the occasional attacks detection aggregated weights $\mathcal{O}_a(s, t_0, t)$ to each trust feedback as shown in Equation 1. $\mathcal{O}_a(s, t_0, t)$ is calculated as follows:

$$\mathcal{O}_a(s, t_0, t) = \frac{\phi * \mathcal{O}_f(s, t_0, t) + \iota * \mathcal{O}_i(s, t_0, t)}{\lambda} \tag{4}$$

where ϕ and $\mathcal{O}_f(s, t_0, t)$ denote the normalized weight of the occasional collusion attacks detection factor and the factor's value respectively. The second part of the equation represents the occasional Sybil attacks detection factor where ι denotes the factor's normalized weight and $\mathcal{O}_i(s, t_0, t)$ denotes the factor's value. λ represents the number of factors used to calculate $\mathcal{O}_a(s, t_0, t)$. For example, if we only consider the occasional collusion attacks detection factor, $\lambda = 1$; if we consider both the occasional collusion attacks detection factor and the occasional Sybil attacks detection factor, $\lambda = 2$.

Change Rate of Trust Metric. To allow TMS to adjust and tweak trust results for cloud services that have been affected by occasional reputation attacks we introduce the change rate of trust factor. The idea behind this factor is to compensate the affected cloud services by the same percentage of damage in the trust results. Given $\mathcal{C}on(s, t_0)$ the conventional model (i.e., calculating the trust results without considering the proposed approach) for a cloud service s in a previous time instance, $\mathcal{C}on(s, t)$ the conventional model for the same cloud service calculated in a more recent time instance, $\mathcal{O}_a(s, t_0, t)$ the occasional attacks detection aggregated weights, and $e_{\mathcal{O}a}$ the occasional attacks percentage threshold. The change rate of trust results factor $\mathcal{C}_t(s, t_0, t)$ is calculated as follows:

$$\mathcal{O}_t(s, t_0, t) = \begin{cases} \left(\frac{\mathcal{C}on(s,t_0)}{\mathcal{C}on(s,t)}\right) - 1 & if \ \mathcal{C}on(s,t) < \mathcal{C}on(s,t_0) \\ & and \ 1 - \mathcal{O}_a(s, t_0, t) \geq e_{\mathcal{O}a} \\ & \\ 0 & otherwise \end{cases} \tag{5}$$

where $\left(\frac{\mathcal{C}on(s,t_0)}{\mathcal{C}on(s,t)}\right) - 1$ represents the change rate of trust results for cloud service s during a period of time $[t_0, t]$. The change rate of trust results will only be used if the conventional model in the more recent time instance is less than the conventional model in the previous time instance and the occasional attacks percentage during the same period of time $[t_0, t]$ (i.e., $1 - \mathcal{O}_a(s, t_0, t)$) is larger or equal to the occasional attacks percentage threshold. For instance, even if the conventional model in the current time for the cloud service a is less than the conventional model 10 days ago, the cloud service a will not be rewarded because the occasional attacks percentage is less than the occasional attacks percentage threshold (e.g., $1 - \mathcal{O}_a(a, t_0, t) = 20\%$ and $e_{\mathcal{O}a} = 30\%$). The change rate of trust results is designed to limit the rewards to cloud services that are affected by slandering attacks [4] (i.e., cloud services that have decreased trust results) because TMS can dilute the increased trust results from self-promoting attacks [3] using the occasional attacks detection factors (i.e., $\mathcal{O}_a(s, t_0, t)$). The adaptive change rate of trust results factor can be used to assign different weights using χ the normalized weight factor as shown in Equation 1.

4 Implementation and Experimental Evaluation

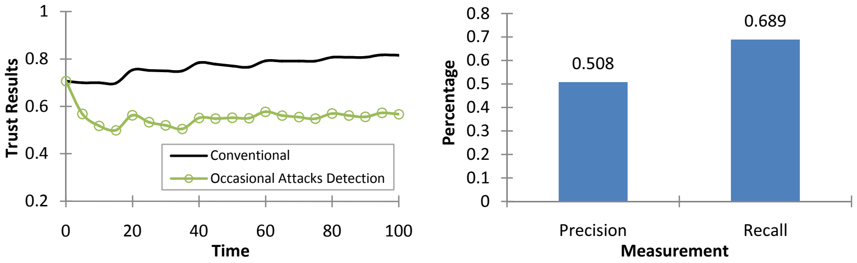
System Architecture. The architecture consists of several layers including: i) *Trust Data Provisioning* for collecting cloud services and trust information where the *Cloud Services Crawler* module is developed based on the Open Source Web Crawler for Java (crawler4j[3]) and extended to allow TMS to automatically discover cloud services on the Internet and the *Trust Feedbacks Collector* module is developed to collect feedbacks directly from consumers and stores them in the *Trust Feedbacks Database*. Moreover, an IdM is developed to allow consumers to establish their identities before using TMS through registering their credentials at the *Trust Identity Registry* where the total number of established identities is collected using the *Identity Info Collector*. ii) *Trust Assessment Function* for handling trust assessment requests from users where the *Factors Calculator* is developed to calculate the occasional attacks detection factors and the *Trust Assessor* to calculate the trust of cloud services by assigning the factors weights to feedbacks and store them in the *Trust Results and Factors Weights Storage*.

[3] `http://code.google.com/p/crawler4j/`

Experimental Design and Setup. In order to validate our approach, we collected real world trust feedbacks on cloud services by crawling review websites such as `CloudHostingReviewer.com` and `cloud-computing.findthebest.com` where consumers usually give their feedback on cloud services that they have used. The collected data is represented in a tuple $\mathcal{H}$ where the feedback represents several QoS parameters as aforementioned and a set of credentials are augmented for each corresponding consumer. We managed to collect 10,076 feedbacks given by 6,982 consumers to 113 real-world cloud services. The collected data is divided into 2 groups of cloud services, one is used to validate our model against occasional collusion attacks and the other is used to validate the model against occasional Sybil attacks. Each cloud service group represents a *Peaks* behavior model. We conducted several experiments to validate the proposed occasional attacks detection model and to demonstrate its robustness against occasional collusion and Sybil attacks. We use two experimental settings: i) measuring the robustness of our model with a conventional model $\mathcal{C}on(s, t_0, t)$ (i.e., turning $\mathcal{O}_a(s, t_0, t)$ to 1 for all feedbacks), and ii) measuring the performance of our model using two measures namely *precision* (i.e., to know how well TMS did in detecting attacks) and *recall* (i.e., to know how many detected attacks are actual attacks). In our experiments, TMS starts rewarding cloud services that have been affected by malicious behaviors when the occasional attacks percentage reaches 25% (i.e., $e_{\mathcal{O}a} = 25\%$), so the rewarding process will occur only when there is a significant damage in the trust result.

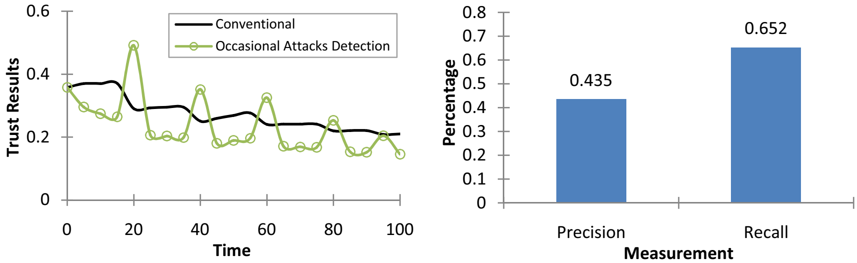
Robustness Against Occasional Collusion Attacks. In occasional collusion attacks experiments, we simulated malicious users to increase trust results of cloud services (i.e., self-promoting attack [3]) by giving multiple feedbacks with the range of [0.8, 1.0]. From Figure 1, we note that results when considering to calculate the trust with our model decrease quickly after a short period of time and the responsible metric for this detection is the occasional collusion attacks detection metric. In addition, we can see that our model achieves 0.508 in precision and scores 0.689 in recall. Overall there is a fair degree in recall which indicates that most of the detected attacks are actual attacks. This means that our model can successfully detect occasional attacks and TMS diluted the increased trust results from self-promoting attacks using the proposed factors.

Robustness Against Occasional Sybil Attacks. In occasional Sybil attacks experiments, we simulated malicious users to decrease trust results of cloud services (i.e., slandering attack [4]) by establishing multiple identities and giving feedbacks with the range of [0, 0.2]. From Figure 2 we can see that trust results when considering to calculate the trust with our model response effectively where 5 peaks in trust results appear (i.e., Figure 2(a)). This is true because the cloud service was rewarded when the occasional attacks occurred. Moreover, we note that the overall precision of our model is 0.435, while the overall recall is 0.652 (See Figure 2(b)). This means that our model can successfully detect occasional Sybil attacks and reward affected cloud services using the change rate of trust factor.



(a) Robustness Against Attacks (b) Attacks Detection

Fig. 1. Occasional Collusion Attacks Experiments



(a) Robustness Against Attacks (b) Attacks Detection

Fig. 2. Occasional Sybil Attacks Experiments

5 Discussions and Conclusion

Over the past few years, trust management has been one of the hot topics especially in the area of cloud computing. Some of the research works use policy-based trust management techniques. For example, Ko et al. [7] proposed TrustCloud framework for accountability and trust in cloud computing which consists of five layers including workflow, data, system, policies and laws, and regulations layers to address accountability in the cloud environment from all aspects. Brandic et al. [1] proposed a novel approach for compliance management in cloud environments to establish trust where the approach is developed using a centralized architecture and uses compliant management technique to establish trust. Unlike previous works that use policy-based techniques, we evaluate the trustworthiness of a cloud service using reputation-based trust management techniques.

Other research works use reputation-based trust management techniques. For instance, Habib et al. [5] proposed a multi-faceted Trust Management (TM) system architecture which models uncertainty of trust information collected from multiple sources using a set of Quality of Service (QoS) attributes such as security, latency, availability, and customer support. Hwang et al. [6] proposed a security-aware cloud architecture where trust negotiation and data coloring techniques are used to support cloud providers and the trust-overlay networks to support consumers. Unlike previous works, we propose an occasional attacks detection model that not only detects misleading trust feedbacks from collusion and Sybil attacks, but also has the ability to adaptively adjust the trust results for cloud services that have been affected by occasional malicious behaviors.

Our work presented in this paper is one of the first few that focuses on the detection of occasional reputation attacks on cloud services. We present several techniques enabling the detection of such attacks. In particular, we introduce an occasional attacks detection model that detects misleading feedbacks from collusion and Sybil attacks. Our model has the capability to adjust trust results for cloud services that have been affected by such malicious behaviors. We also have collected a large collection of consumer's trust feedbacks given on real-world cloud services to evaluate and demonstrate the applicability of our approach.

Acknowledgments. Talal H. Noor and Abdullah Alfazi's work has been supported by King Abdullah's Postgraduate Scholarships, the Ministry of Higher Education, Kingdom of Saudi Arabia.

References

1. Brandic, I., et al.: Compliant Cloud Computing (C3): Architecture and Language Support for User-Driven Compliance Management in Clouds. In: Proc. of CLOUD 2010 (2010)
2. David, O., Jaquet, C.: Trust and Identification in the Light of Virtual Persons (June 2009), `http://www.fidis.net/resources/deliverables/identity-of-identity/` (accessed March 10, 2011)
3. Douceur, J.R.: The Sybil Attack. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 251–260. Springer, Heidelberg (2002)
4. Friedman, E., et al.: Manipulation-Resistant Reputation Systems. In: Algorithmic Game Theory, chap, pp. 677–697. Cambridge University Press, New York (2007)
5. Habib, S., et al.: Towards a Trust Management System for Cloud Computing. In: Proc. of TrustCom 2011 (2011)
6. Hwang, K., Li, D.: Trusted Cloud Computing with Secure Resources and Data Coloring. IEEE Internet Computing 14(5), 14–22 (2010)
7. Ko, R., et al.: TrustCloud: A Framework for Accountability and Trust in Cloud Computing. In: Proc. of SERVICES 2011 (2011)
8. Noor, T.H., Sheng, Q.Z.: Credibility-Based Trust Management for Services in Cloud Environments. In: Kappel, G., Maamar, Z., Motahari-Nezhad, H.R. (eds.) ICSOC 2011. LNCS, vol. 7084, pp. 328–343. Springer, Heidelberg (2011)
9. Noor, T.H., Sheng, Q.Z.: Trust as a Service: A Framework for Trust Management in Cloud Environments. In: Bouguettaya, A., Hauswirth, M., Liu, L. (eds.) WISE 2011. LNCS, vol. 6997, pp. 314–321. Springer, Heidelberg (2011)
10. Ren, K., et al.: Security Challenges for the Public Cloud. IEEE Internet Computing 16(1), 69–73 (2012)