Advances in Intelligent Systems Research, volume 159
International Conference on Mathematics, Modelling, Simulation and Algorithms (MMSA 2018)

User Trust for Access Control in Software Defined Networking

Chanchan Zhao[1,2,*] and Feng Liu[1]
[1]School of Computer and Information Technology, Beijing Jiaotong University, Beijing, 100044, P.R. China
[2]College of Information Engineering, Inner Mongolia University of Technology, Hohhot, 010051, P.R. China
[*]Corresponding author

***Abstract*—The proposition of increased innovation in network applications and reduced cost for network operators has won over the networking world to the vision of Software-Defined Networking (SDN). In this new architecture, the network resources at the data plane are shared and different tenants can have competing objectives. Therefore, mechanisms are needed to protect the network resources from unauthorized access. Considering the security requirements and resource consumption, this study presents an user trust method for more finely granular access control. It is able to make control decisions in response to the different user and resources. Rigorous analysis and extensive simulations have demonstrated its distinguished performance in terms of user identity identification, user rights adjustment and effectively denied malicious access.**

Keywords—user trust; access control; Software-Defined Networking

I. INTRODUCTION

Today Internet-based systems, such as cloud services and social networks, change their network requirements (e.g., bandwidth demand, topology, and routing information) dynamically. However, that means managing and controlling network traffic is undergoing a major change. In recent years, this problem has been addressed by researchers through Software Defined Networking (SDN) which has emerged as a new network architecture that allows for more flexibility through software-enabled network control [1, 2]. At the core of SDN is the separation between the network's data plane and its control plane. The control plane comprises a logically centralized software entity—the controller—running on commodity hardware[3, 4].

There have been many attempts to make networks more manageable and more secure. Note that by using network virtualization [5, 6] the network owner can provide an isolated view on the network to each leasing tenant. However, the different tenants still share the data plane's network resources. Meanwhile，without security function, all SDN applications have full access to the underlying network enabling the possibility for potential malicious applications. Thus, SDN controller should ensure that the sensitive information should be offered only to the ones who are authorized, but in fact open APIs can offer the common information to the public.

To solve the issue of illegal access, a user trust method in the software defined networking was proposed. Obtain the user's trust according to the calculation and updating method of the user trust. The experimental results showed the user trust was correct and effective, and the user's real identity could be reflected in the scheme through trust, and it had a better ability to recognize the malicious illegal users to effectively prevent illegal access and protect lawful access.

The remainder of this paper is organized as follows. Section II reviews some related studies. In Section III, we propose an user trust in the software defined networking. The analysis is further validated by extensive simulation experiments introduced in Section IV. Finally, Section V concludes this research work.

II. RELATED WORK

Current network attacks have been coming from not only outside of an organization but also internal networks in recent years due to malware infected clients and malicious insiders[7-9]. Existing researches have revealed the vulnerability of the SDN controller to attack[10]. Hence, research efforts have been focused on how to effectively apply minimum privilege on the applications protecting the network from control-plane attacks. Further, both authentication and authorization of the applications is required to ensure that only trusted applications should connect to the network. Yu et al. present a secure SDN structure with each network element managed by multiple controllers using the Byzantine mechanism[11]. In a similar approach, Othman et al. present a signature algorithm to securely transmit flow installation requests from network device to network device. The system requires a centralized trust manager and introduces significant overhead in message-passing and signature checking[12]. One interesting issue is that exposing the full privilege of OpenFlow to every application without protection is identified. Existing schemes propose PermOF [7] with 18 permissions and an isolation mechanism to enforce the permissions at the Application Programming Interface (API) entry. This approach is only effective as long as the applications reside in the controller and fails to address the security challenges that arise while applications are deployed external to the controller. The concept of the permissions system is extended in [13]. OperationCheckpoint is designed and implemented on the Floodlight controller. Scott-Hayward et al. define the set of permissions to which the application must subscribe on initialization with the controller and introduce an OperationCheckpoint, which implements a permissions check

prior to authorizing application commands. This approach, however, is controller-dependent and also lacks a mechanism to verify the authenticity of external applications. To deny access to the SDN by unauthorized hosts, Auth-Flow, an authentication and access control mechanism based on host credentials, is proposed in [14]. The controller allows or denies traffic based on the authentication response. Access control is implemented by pairing host credentials with a set of host flows. However, these work take the scenarios less involve user authentication. The user's identity authentication should be able to solve the problem of measuring whether the user's behavior is legal and whether the abnormal behavior can be stopped. In this paper, we argue that a new method of user trust that addresses safe access control issues with consideration on efficiency.

III. USER TRUST

Generally, a new user can get a legal identity and further get the corresponding privilege to the resources. However, the network may include a potential security problem described as follows: To be trusted, a malicious user will first disguise as a legal user, successfully register and get the authorization to the corresponding resources, try to access unauthorized resources, and perform illegal access. In addition, with elapse of time, the original legal user may temporarily try to access unauthorized resources. Therefore, the user behaviors shall be monitored in real time to identify normal users and malicious users and avoid illegal access. To solve the above problem, the concept of the user trust is introduced. The trust indicates the trust degree of network resources or network services to users, which evaluates attributes and behaviors among entities. This paper regards the user trust as a key basis to identify user identities and adjust user privileges. For easy further expatiation, the users, services (resources) and roles in the software defined networking are described in a formal manner. U_{set} indicates the user set in the network and $U_{set}=\{user_1, user_2, \cdots\}$. S_{set} indicates the set of services or resources in the network and $S_{set}=\{service_1, service_2, \cdots\}$. $Role_{set}$ indicates the set of user roles and $Role_{set}=\{role_1, role_2, \cdots\}$.

First, the influences of the history trust on the current trust will change with time. With elapse of time, the reference meaning of the history trust to the current trust degree will become smaller and smaller. The longer trust has smaller reference meaning. To reflect such dynamic change, one time decay function $\delta(t)$ is defined, which indicates the decay of the influences of the history trust to the current trust with elapse of time. Provided that the shorter time to this interaction indicates quicker time decay and the longer time to this interaction indicates slow time decay. Based on this assumption, the following equation can be obtained:

$$\frac{d\delta(t)}{dt}=-\theta\delta(t) \tag{1}$$

Based on the equation (1), the time decay function $\delta(t)$ can be represented as follows:

$$\delta(t)=Ce^{-\theta t} \tag{2}$$

θ is the time decay regulation parameter and is set according to actual conditions. Generally $\theta=\frac{Ln2}{T_c}$, T_c is the trust half-life. After a T_c, the trust influence will reduce by half.

In addition, to improve the trust, a malicious user may perform well in case of acquisition of common services and perform dishonestly in case of acquisition of critical services. When the user trust is calculated, the importance of the interactive services will be also analyzed. To represent the influences of the service importance on the user trust, here the interaction factor is proposed and is normalized to [0, 1]. Bigger interaction factor indicates higher interaction importance and bigger influences on the user trust. On the contrary, a smaller interaction factor indicates lower interaction importance and smaller influences on user trust.

E.g. the interaction factor of key data will be bigger than the common data. One typical example is described as follows: In the P2P transaction, if the amount of the transaction fund is x, the interaction factor is calculated as follows:

$$f(x)=\begin{cases}\sqrt{(\frac{x}{M_0})}, & x<M_0\\ 1, & x\geq M_0\end{cases} \tag{3}$$

M_0 is the threshold of the fund amount. When the transaction amount is bigger than M_0, the value of the interaction factor is 1.

Based on the above analysis on the time decay and interaction factor, the UT_{ij} of the user is defined as follows:

$$UT_{ij}(user_i, service_j)=\begin{cases}0.5 & n=0\\ \frac{\sum_{r=1}^{n} UT_{ij}^{r}\delta(t_r)InF_r}{\sum_{r=1}^{n}\delta(t_r)InF_r} & n>0\end{cases} \tag{4}$$

r is the interaction SN of the $user_i$ and $service_j$. n is the total completed interactions of $user_i$ and $service_j$. $n=0$ indicates that $user_i$ accesses the $service_j$ for the first time. If the initial value 0.5 is given, it indicates one medium trust. UT_{ij}^{r} is the trust after rth interaction. $\delta(t_r)$ is the time decay function when the difference between this interaction SN and a history interaction SN is t_r. InF_r is the service influence factor

in case of r[th] interaction between $user_i$ and $service_j$, $InF_r \in [0,1]$.

Based on the above analysis, the basic framework of the user trust constructed, as shown in Figure I.

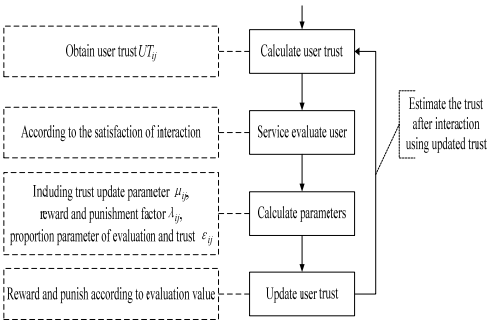

FIGURE I. THE BASIC FRAMEWORK OF TRUST.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

This section mainly describes three simulation tests: The first test gives the simulation results of the access control scheme. The second test studies the influences of parameters on the user trust, including influence analysis of the time decay regulation parameter and the evaluation-trust ratio on the trust. The third test studies the influences of the user attributes number on the encryption algorithm performance.

A. Simulation Results

The following scenario is given: provided that $user_A$, $user_B$ and $user_C$ exist. $user_A$ is the trusted user. After each interaction, the user evaluation value SE_{ij} for the service is 1 and $user_B$ is a general user. After each interaction, the user evaluation value SE_{ij} is 0.6 or 0.5 for the service. $user_C$ is a malicious user. After each interaction, the user evaluation value SE_{ij} is 0.3 or 0.1 for the service. Three users perform 10 interactions. To reduce calculation difficulty of the time decay function, the time t is calculated by the user and service interaction number under the premise of no influences on user trust accuracy. After three users interact with the service, the user trust is shown in Figure II. In this figure, the x coordinate indicates the interaction number between the users and services. The y coordinate indicates the calculated trust after each interaction between the user and service.

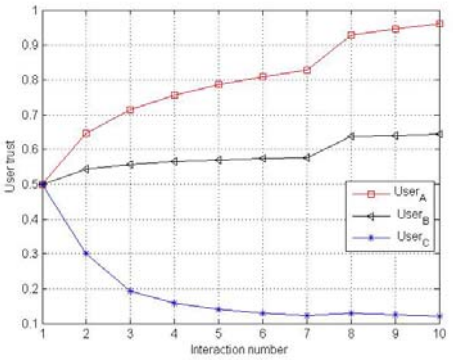

FIGURE II. USER TRUST.

In addition, multiple users are simulated, including 200 trusted users, 50 general users and 50 malicious users. After the user trust is calculated, the users are classified into three types by trust and the mean of each class is calculated. The simulation test results are shown in Figure III.

As shown in Figure III, for trusted users, the user trust value is between 0.9 and 1 and indicates that the trusted users are reliable all the time. For general users, the trust is within the reasonable interval. For malicious users, the trust will reduce and identities of malicious users become more specific with growth of interactions.

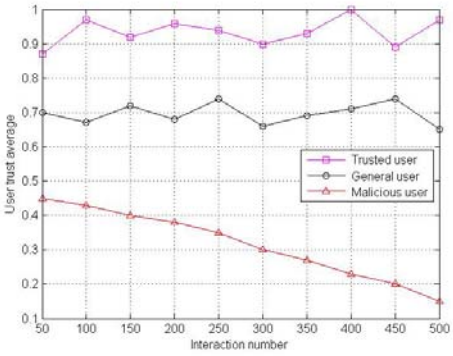

FIGURE III. USER TRUST AVERAGE.

B. Parameters Influence on Trust

If $T_c = 5$ is the trust half-life, then $\theta = \frac{Ln2}{5}$, time decay function is shown in Figure IV.

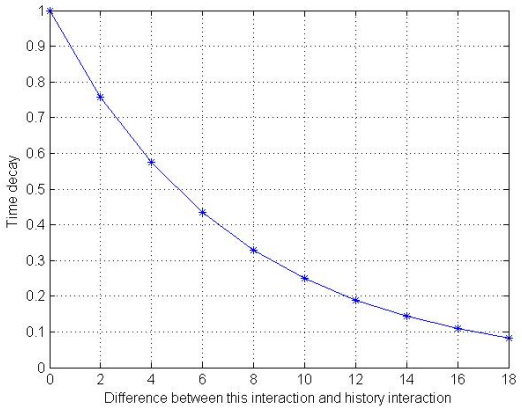

FIGURE IV. TIME DECAY.

As shown in Figure IV, the time decay will reduce with growth of the difference between this interaction SN and the history interaction SN. When the difference increases to a value, the time decay will approximate to the zero. Such change complies with the trust theory and proves correctness of the equation 2[15]. The decay parameter θ can be adjusted to control the speed of the time decay change and regulate the change rate of the trust. E.g. when a malicious behavior occurs, a bigger θ can be set to quickly reduce the time decay and further quickly reduce the user trust. The Figure V gives the change of the time decay for different θ.

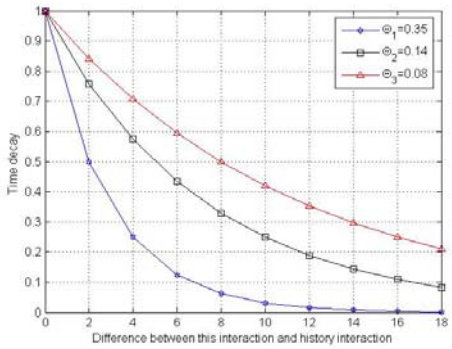

FIGURE V. DIFFERENT θ.

V. CONCLUSION

In this paper, we investigated the user trust in the software defined networking. We formulated the user trust problem as an identification of user identity, which can adjust user rights while requirements of user access resource changes in real time. This method is easy to be implemented on line, and can achieve more finely granular access control.

ACKNOWLEDGMENT

This work is supported by the National Key R&D Plan (No. K17B500071) and the Science and Technology R&D Plan of China Railway Corporation (No. 2016J007-B).

REFERENCES

[1] B. Raghavan, M. Casado, T. Koponen et al. “Software-defined internet architecture: decoupling architecture from infrastructure”, ACM Workshop on Hot Topics in Networks, vol. 8, no. 4, pp. 43-48, 2012.

[2] T. Simonite. “MIT Technology Review Announces 10 Breakthrough Technologies”, Sensors, 2015.

[3] P. Porras, S. Cheung, M. Fong, et al. “Securing the Software Defined Network Control Layer”, in Proc. of Network and Distributed System Security Symposium, pp. 1-15, 2015.

[4] H. Z. Wang, P. Zhang, L. Xiong, et al. “A secure and high-performance multi-controller architecture for software-defined networking”, Frontiers of Information Technology & Electronic Engineering, vol. 17, no. 7, pp. 634-646, 2016.

[5] M. Casado, T. Koponen, R. Ramanathan, et al. Virtualizing the network forwarding plane, 2010.

[6] R. Sherwood, G. Gibb, K. K. Yap, et al. “Can the production network be the testbed?”, in Proc. of Usenix Symposium on Operating Systems Design and Implementation, pp. 365-378, October 4-6, 2010.

[7] X. Wen, Y. Chen, C. Hu, et al. “Towards a secure controller platform for openflow applications”, in Proc. of 2nd ACM SIGCOMM workshop on Hot topics in software defined networking, pp. 171-172, 2013.

[8] S. Scott-Hayward, S. Natarajan and S. Sezer. “A Survey of Security in Software Defined Networks”, IEEE Communications Surveys & Tutorials, vol. 18, no. 1, pp. 623-654, 2016.

[9] S. Lee, C. Yoon, C. Lee, et al. “DELTA: A Security Assessment Framework for Software-Defined Networks”, in Proc. of Network and Distributed System Security Symposium, 26 February - 1 March, 2017.

[10] H. Li, P. Li, S. Guo, et al. “Byzantine-resilient secure software-defined networks with multiple controllers”, in Proc. of IEEE Int.Conf. on Communications, pp. 695-700, 2014.

[11] D. Yu, A. W. Moore, C. Hall, et al. “Authentication for Resilience: The Case of SDN”, ser. Security Protocols XXI. Springer, vol. 8263, pp. 39-44, 2013.

[12] O. Othman and K. Okamura. “Securing Distributed Control of Software Defined Networks”, International Journal of Computer Science & Network Security, vol. 13, no. 9, pp. 5-14, 2013.

[13] S. Scott-Hayward, C. Kane and S. Sezer. “OperationCheckpoint: SDN Application Control”, in Proc. of International Conf. on Network Protocols.pp. 618-623, 19 October , 2014.

[14] D. M. F. Mattos and O. C. M. B. Duarte. “AuthFlow: authentication and access control mechanism for software defined networking”, Annals of Telecommunications, vol. 71, no. 11-12, pp. 1-9, 2014.

[15] T. Grandison and M. Sloman. “A survey of trust in internet applications”, IEEE Communications Surveys & Tutorials, vol. 3, no. 4, pp. 2-16, 2009.