Table 5: Configuration of the Formal Access Control Model defined in Table 2 for the Use Case Scenario in Section 5.

$A = \{LS, LB, NIP, FW, OC\}$,
$R = \{APP, SEC, ADMIN\}$ with a total order > on R ,as defined in Table 2,
$T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{15}, t_{16}, t_{17}, t_{18}\}$, as labled in Table 1,
$AR = \{(LS, APP), (LB, APP), (NIP, SEC), (FW, SEC), (OC, ADMIN)\}$,
$TR = \{(t_i, APP), (t_{13}, SEC), (t_j, ADMIN) \mid (t_i \in T \mid 1 \leq i \leq 12, t_j \in T \mid 14 \leq j \leq 18))$,
$DXOP = \{'add\ flow\ rule', 'packet\ in', 'flow\ stats', 'packet\ out'\}$,
$Type('add\ flow\ rule') = '\ Flow\ rule\ mod', Type('packet\ in') = '\ Packet - In\ return',$
$Type('flow\ stats') = '\ Switch\ stats\ request' = '\ Switch\ stats\ report', Type('packet\ out') = '\ Packet - Out',$
$AuthorizationRule(LS, '\ add\ flow\ rule') = true, AuthorizationRule(LB, '\ add\ flow\ rule') = true,$
$AuthorizationRule(FW, '\ add\ flow\ rule') = true,$
$AuthorizationRule(LS, '\ packet\ in') = true, AuthorizationRule(LB, '\ packet\ in') = true, AuthorizationRule(NIP, '\ packet\ in') = true,$
$AuthorizationRule(FW, '\ packet\ in') = true\ AuthorizationRule(OC, '\ packet\ in') = true,$
$AuthorizationRule(LB, '\ flow\ stats') = true, AuthorizationRule(FW, '\ packet\ out') = true.$

apps and data in the SDN controller. Their system uses API hooking to intercept the app execution flow to protect the controller. Prior to SE-Floodlight, FortNOX [22] implements a role-based authorization with three roles. SE-Floodlight [9] is an extension and improvement of the FortNOX. Tseng et al [15], inspired by [9], proposed Controller-DAC with API requests threshold and a priority for each app assigned either directly or via the role.

SM-ONOS [5] proposed a permission system at four-level granularity. First, code packages are classified as either app or non-app OSGi bundles. Next, app bundles are assigned either admin or user role with the appropriate permissions. Non-administrative API-permissions then granted to apps followed by network-level permissions. Based on API-level permissions from SM-ONOS, [3] proposed information flow control among apps for the ONOS controller.

8 CONCLUSION AND FUTURE WORK

In this paper, we formalized a role-based authorization model for SDN using SE-Floodlight as a reference controller and proposed an administration model. Then we showed a configuration of the formal model for a use case scenario. We then discussed the security aspects of the authorization model and described some security problems related to over-privileged apps, limitations of role hierarchy, app upgrading, and app downgrading problems. Finally, we proposed a solution to overcome the mentioned problems.

As a future work, first, we plan to design an access control model that includes a wider set of SDN operations. Second, we plan to present a sophisticated access control model using RBAC standard and Attribute-based access control (ABAC) terminology in order to achieve a fine-grained access control within a holistic view to resources in SDN environment.

ACKNOWLEDGMENTS

This work is partially supported by NSF CREST Grant HRD-1736209 and DoD ARL Grant W911NF-15-1-0518.

REFERENCES

[1] Ijaz Ahmad, Suneth Namal, Mika Ylianttila, and Andrei Gurtov. 2015. Security in software defined networks: A survey. *IEEE Communications Surveys & Tutorials* 17, 4 (2015), 2317–2346.
[2] Scott-Hayward el al. 2014. Operationcheckpoint: Sdn application control. In *Network Protocols (ICNP), 2014 IEEE 22nd International Conference on*. IEEE.
[3] B. Ujcich et al. 2018. Cross-App Poisoning in Software-Defined Networking. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 648–663.
[4] C. Banse et al. 2015. A secure northbound interface for sdn applications. In *Trustcom/BigDataSE/ISPA, 2015 IEEE*, Vol. 1. IEEE, 834–839.
[5] C. Yoon et al. 2017. A Security-Mode for Carrier-Grade SDN Controllers. In *Proceedings of the 33rd Annual Computer Security Applications Conference*. ACM.
[6] D. Kreutz et al. 2013. Towards secure and dependable software-defined networks. In *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*. ACM, 55–60.
[7] H. Padekar et al. 2016. Enabling dynamic access control for controller applications in software-defined networks. In *Proceedings of the 21st ACM on Symposium on Access Control Models and Technologies*. ACM, 51–61.
[8] J. Noh et al. 2016. Vulnerabilities of network OS and mitigation with state-based permission system. *Security and Communication Networks* 9, 13 (2016).
[9] Porras P. A et al. 2015. Securing the Software Defined Network Control Layer.. In *NDSS*.
[10] R. Sherwood et al. 2010. Can the production network be the testbed?. In *OSDI*.
[11] Scott-Hayward et al. 2013. SDN security: A survey. In *Future Networks and Services (SDN4FNS), 2013 IEEE SDN For*. IEEE, 1–7.
[12] Scott-Hayward et al. 2016. A survey of security in software defined networks. *IEEE Communications Surveys & Tutorials* 18, 1 (2016), 623–654.
[13] X. Wen et al. 2013. Towards a secure controller platform for openflow applications. In *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*. ACM, 171–172.
[14] X. Wen et al. 2016. Sdnshield: Reconciliating configurable application permissions for sdn app markets. In *Dependable Systems and Networks (DSN), 2016 46th Annual IEEE/IFIP International Conference on*. IEEE, 121–132.
[15] Y. Tseng et al. 2017. Controller DAC: Securing SDN controller with dynamic access control. In *Communications (ICC), IEEE International Conference on*. IEEE.
[16] David F Ferraiolo, Ravi Sandhu, Serban Gavrila, D Richard Kuhn, and Ramaswamy Chandramouli. 2001. Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security (TISSEC)* 4, 3 (2001), 224–274.
[17] Security Enhanced Floodlight. 2018. https://www.sdxcentral.com/projects/openflow-sec-security-enhanced-floodlight/.
[18] Floodlight-Project. 2018. http://www.projectfloodlight.org/.
[19] Ryu SDN Framework. 2018. http://osrg.github.io/ryu/.
[20] ON.Lab. ONOS. 2018. http://onosproject.org/.
[21] The OpenDaylight platform. 2018. https://www.opendaylight.org/.
[22] Philip Porras, Seungwon Shin, Vinod Yegneswaran, Martin Fong, Mabry Tyson, and Guofei Gu. 2012. A security enforcement kernel for OpenFlow networks. In *Proceedings of the first workshop on Hot topics in software defined networks*. ACM.
[23] Ravi S Sandhu, Edward J Coyne, Hal L Feinstein, and Charles E Youman. 1996. Role-based access control models. *Computer* 29, 2 (1996), 38–47.
[24] Changhoon Yoon, Taejune Park, Seungsoo Lee, Heedo Kang, Seungwon Shin, and Zonghua Zhang. 2015. Enabling security functions with SDN: A feasibility study. *Computer Networks* 85 (2015), 19–35.