DREAD-R: Severity Assessment of ONOS SDN Controller

Muhammad Shakil, Alaelddin Fuad Yousif Mohammed(✉), Hyeontaek Oh, and Jun Kyun Choi

Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea
{shakilphd,alaelddin,hyeontaek}@kaist.ac.kr, jkchoi59@kaist.edu

Abstract. In few past years, popularity of Software Defined Networking (SDN) among academia and industry is rapidly increased, and users are conferenced about choosing suited and secured SDN controller. Recently, Open Network Operating System (ONOS), which provides the control plane for SDN, appears as best choice for service provider in term of high availability, scalability, and security. There are some existing models for security assessment of SDN. However, there is still a room for more assessments. This paper address the severity assessment of ONOS using proposed DREAD-R model which considers traditional DREAD (Damage potential, Reproducibility, Exploitability, Affected users and Discoverability) model with additional "Reputation" parameter. This paper found that control plane vulnerabilities are critical in nature and disrupt entire network functions and need immediate attention for solutions.

Keywords: SDN · ONOS · Severity assessment · DREAD-R

1 Introduction

Software Defined Networking (SDN) is a novel network approach that separates control plane from forwarding plane. SDN provides several advantages over the legacy network such as centralized network visualizing, the agility of control plane programing, and central security implementation [1–3]. We can notice from academia and industry that SDN is a promising and disruptive network technology to solve legacy network challenges.

In the legacy network, control plane and data plane are packaged together. However, in SDN, data plane is separated from control plane [4]. The separation of control plane from data plane provides a global view of the network. Network performance can be viewed and managed centrally from SDN controller (i.e. it is the mainstay of SDN innovation).

SDN architecture can be viewed as a combination of three layers as shown in Fig. 1. On the top is the application layer which, contains network functions. Therefore, all kinds of applications remain in this layer (e.g. firewalls, load balancing, traffic monitoring). The SDN applications communicate to a controller through northbound interface [4]. Many of the northbound Application program

S. Phon-Amnuaisuk et al. (Eds.): MIWAI 2017, LNAI 10607, pp. 323–330, 2017.
https://doi.org/10.1007/978-3-319-69456-6_27

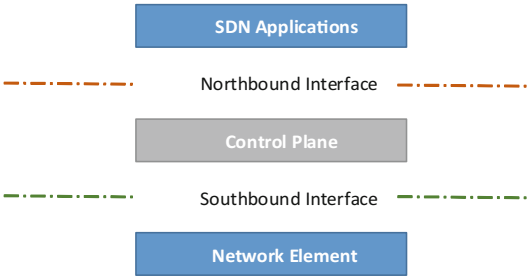

Fig. 1. SDN architecture.

interfaces (APIs) that exist are based on their applications. The commonly used APIs and protocols that communicate between application and controller are Representational State Transfer (REST) Application Program Interface (API). On the other hand, network element communicates using southbound interface with the controller. Southbound protocols are also open to adopt any protocol according to their needs (e.g. Network Configuration Protocol (NETCONF), Open vSwitch Database (OVSDB), and OpenFlow). However, OpenFlow, which is widely used, has become a de facto standard for SDN.

In SDN, the SDN controller is the brain of the network and the owner of virtual resources [4]. Control plane is also responsible for coordinate between network and virtual resources. Usually, a controller provides a concrete interface to the network, and the controller can have different types of access levels to users, devices, and applications [5]. The controller is the center-of-gravity in SDN concept, and it should be resilient against known and unknown attacks [2,6–9]. The controller has multiple north, south, and east/west bound interfaces which interact with applications, network elements as well as with other SDN controllers, respectively. The vulnerabilities of interfaces can be used to compromise the controller [4].

Open Network Operating System (ONOS) is one of the candidates for distributed SDN Controller which has been widely adopted in academia and industry [10]. Security of ONOS has been assessed in some recent studies (e.g. [5,7,11]). Authors in [5] assessed the security of ONOS using Microsoft developed threat modeling technique STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service (DoS), and Elevation of Privilege [12]) using Data Flow Diagramming (DFD). On the other hand, authors in [7] developed a security framework for SDN vulnerability assessment (named DELTA) using fuzzing techniques. Microsoft developed DREAD (Damage potential, Reproducibility, Exploitability, Affected users and Discoverability) risk assessment model to measure the severity of threats/attacks to find out severity [11]. Those previous studies concentrated on analyzing general ONOS security analysis.

In this paper, we assess the severity of threats/attacks utilizing the existing DREAD model with additional proposed parameter "Reputation" to test the

trust of ONOS community. Furthermore, threats/attacks are categorized using the proposed DREAD-R model to find out the severity of SDN controller particularly, ONOS. We rate threats found by DELTA. It has to note here that DELTA is the first ever SDN security analysis framework (also known as SDN pentesting framework) [7] and STRIDE based assessment [5]. In this paper, we are interested in analyzing the security vulnerabilities/threats found using DELTA and STRIDE and rating them using proposed DREAD-R model. The goal of this paper is to check the validity of the following hypotheses:

- H1. Control plane vulnerabilities are more severe than northbound and southbound protocols;
- H2. Proposed DREAD-R model increase/decrease severity rating based on reputation.

The rest of this paper is organized as follows, Sect. 2 describes general ONOS architecture and literature review of STRIDE, DELTA and DREAD security assessment frameworks. Section 3 introduces proposed DREAR-R and the analysis of the severity assessment of ONOS. Section 4 presents our conclusion and future work.

2 Related Work

2.1 ONOS

ONOS is an open source distributed network operating system for service providers. ONOS core built on carrier grade features that provide high availability, scale out, and performance. It also provides abstractions to Northbound and Southbound interfaces [10] as shown in Fig. 2. ONOS is a JAVA based controller and uses Open Service Gateway Initiatives (OSGI) to develop different subsystems and provide web style agility to SDN control plane.

Commonly, ONOS releases do not provide any built-in security mechanism. However, a separate implementation released with the name of Security-Mode ONOS. In this study, we analyze ONOS Security-Mode that is referred as a conservative mode. The conservative mode has two features; (1) Application Authentication and (2) Role based/Permission based Access Control (least privilege applications) [5].

2.2 Security Assessment Frameworks for ONOS

STRIDE: STRIDE stands for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service (DoS), and Elevation of Privilege [12]. Spoofing means misleading victim and attract them to make improper security decision [13]. Tampering includes alteration of data. Repudiation involves denying of action or lack of the ability of tracing the forbidden actions. Information disclosure involves revelation of data to unauthorized individuals. If a valid user cannot access any service then it is DoS attack. In Elevation of threats, an ordinary user

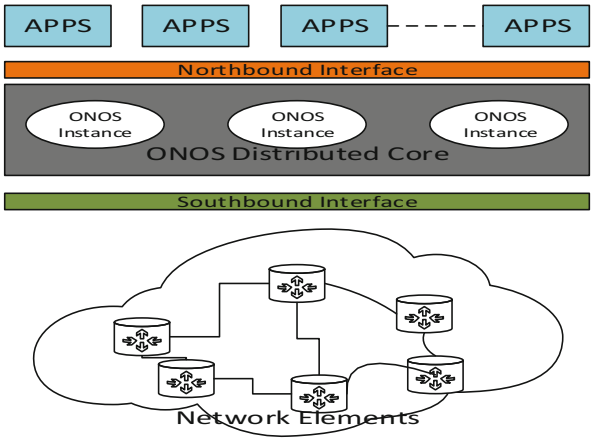

Fig. 2. ONOS architecture

can have super or privileged user access that supposed not to have [12]. STRIDE is a model-based technique for threat modeling and required Data Flow Diagram (DFD) [13]. DFD is visual diagram to describe logical model that defines data transformation in system without operation sequence. DFD defines scope of the security analysis and produce a model of underneath system. Authors in [5] used STRIDE model for ONOS controller analysis where. In our paper, we utilize threats that explored in [5] to rank risks. Furthermore, several threat modeling are available for implementation such as Trike [14], Secure UML [15], Flexible Modeling Framework (FMF) [16], and PASTA [17].

DELTA: SDN security evaluation framework DELTA developed by authors in [7] is the only available open source penetration testing tool to assess SDN security using fuzzing techniques and found known attacks as well as unknown attacks. Known attacks of DELTA targets SDN control flow based on the well-known protocol OpenFlow, dividing them into three categories such as symmetric flows, asymmetric control flows, and intra-controller control flows [7]. Numbers of SDN security hypothetical analysis work has been done in previous work (e.g. [18–20]). DELTA is the only empirical penetration framework based on SDN controller that includes ONOS for evaluation along with other SDN controllers [7]. Therefore, this is motivate us to use DELTA in our study. Additionally, DELTA also exposed unknown attacks that are not part of this study. In this study, we used only known attacks found authors in [7] and rate them according to proposed DREAD-R model.

DREAD: DREAD is a popular model to rank/priorities risks to related threats/attacks in numeric values. DREAD is acronym of Damage potential (How much is the damage if the vulnerability is exploited?), Reproducibility

(How easy is it to reproduce the attack?), Exploitability (How difficult is to exploit vulnerability?), Affected users (As a rough percentage, how many users are affected?), and Discoverability (How easy is it to find the vulnerability?) [11]. Also, Common Vulnerability Scoring System (CVSS) can be an alternative to DREAD for measuring the severity of threats [21].

3 Proposed DREAD-R Severity Assessment Modeling and Analysis

3.1 Proposed Model

This paper adds "Reputation" criterion (How much reputation of ONOS project/community is damage?) to DREAD model for measuring trust effects on ONOS community. If ONOS core is affected, that is categorized as high risk in damage potential, however, if an attacker can leak trivial information from third party application or ONOS subsystem, this can be categorized as low risk. An attacker can reproduce ONOS attack without any time window (whenever attacker want), this is categorized as high risk. On the other hand, an attacker may have knowledge of security loop hole and ONOS architecture, but it is still difficult to mount a similar attack, this is categorized as low in

Table 1. Proposed DREAD-R criteria for severity assessment

	Rating	High (3)	Medium (2)	Low (1)
D	Damage potential	Attacker can gain ONOS core access as privileged user, can change core code as well as exploit Apps and install malicious apps	Attacker can leak apps as well as ONOS core information and can disconnect underlying network elements or cause disruption of apps to controller	Attacker can have trivial information of others applications
R	Reproducibility	ONOS user can trivially exploit vulnerability without any time windows	Required few steps to exploit in particular case or in particular time	The attack cannot reproduce even with knowledge of the security hole
E	Exploitability	Attacker having knowledge little knowledge of SDN protocol or app can exploit	A skilled programmer having knowledge of ONOS could make the attack, and then repeat the steps	The attack requires an extremely skilled person and in-depth knowledge of ONOS as well as protocol, every time to exploit
A	Affected users/ Apps/modules	All apps/modules of ONOS/users, default configuration may be affected, key customers	Specific apps/modules of ONOS/users affected or underlying network element	Some of apps/users affected
D	Discoverability	Attack available publically and can easily found using search engines and is part of ONOS core features	The vulnerability is not part of ONOS core and only affect some protocol/North/South bound interface	Can be found by monitoring ONOSE core and apps
R	Reputation	Having Impact on repute of ONOS project and loss of user's and community trust	Having Impact on users trust but not all community	Having no impact on trust of user or ONOS community

term of "Reproducibility". Reputation is widely used among research community [22,23]. However, "Reputation" parameter used in our work is classified as the impact on ONOS user and community trust, which could damage ONOS reputation. Details of all DREAD-R threats categorization correspondence to ONOS summarized in Table 1.

3.2 Result Analysis

A simple scheme of threat rating as high (3), medium (2), and low (1) used as shown in Table 2. Threat ranked on a scale of 1 to 3 where higher number shows higher threat level. After obtaining ratings of individual threat, a sum of DREAD-R presented at the end. Over all rank determined by dividing threats into three categories. Rating of 0–6 as trivial (can be fixed in next release or later), 7–12 as near critical (n-critical) (need to be fixed in next release or earlier) and 13–18 as critical (need to be fixed as early as possible) as shown in Table 2.

Table 2. Severity ranking based on the DREAD-R

Threat studies	Threat detail		D	R	E	A	D	R	Total	Severity
Ramachandra/ STRIDE [5]	**Component**	**Threats**								
	South bound interface	Denial of service	2	2	2	2	2	2	**12**	n-critical
	Process controller core	Denial of service	3	2	2	3	3	3	**16**	critical
	Switches and north bound applications	Spoofing	1	2	1	1	1	0	**6**	trivial
DELTA [7]	**Flow type**	**Attacks**								
	Symmetric flows	Control message manipulation	3	3	1	3	2	2	**14**	critical
	Asymmetric	Control message drop	3	2	1	2	2	2	**12**	n-critical
	Flows	Control message infinite loop	3	2	1	3	2	2	**13**	critical
	Intra-controller control flows	PACKET IN flooding	1	2	2	1	2	1	**9**	n-critical
		Flow rule flooding	2	2	1	1	2	1	**9**	n-critical
		Flow rule modification	2	2	1	2	2	1	**10**	n-critical
		Switch firmware misuse	2	2	1	2	2	1	**10**	n-critical
		Flow table clearance	2	2	1	2	2	1	**10**	n-critical
		Eavesdrop	1	2	1	1	1	0	**6**	trivial
		Man-in-the-middle	3	2	1	3	2	2	**13**	critical
		Internal storage misuse	3	3	1	3	3	3	**16**	critical
		Application eviction	2	2	1	2	3	2	**12**	n-critical
	Non flow operations	System command execution	3	3	2	3	3	3	**17**	critical
		Memory exhaustion	2	2	1	3	2	2	**12**	n-critical
		CPU exhaustion	2	2	1	3	2	2	**12**	n-critical

Denial of Service (DoS) attack through ONOS southbound or data plane rated as critical. DoS attack can disrupt network element communication to ONOS core. ONOS core can be accessed by any of interface, this needs to be fixed in the next release or as early as possible. Spoofing attack on switches or application to leak information from them categorized as trivial and need to be

fixed in the next release or later. Details of threats and attacks categorized are summarized in Table 2.

For severity assessment, 6 threats and attacks are critical and 5 out of 6 are using control plane vulnerabilities to exploit. Attacks affecting northbound and southbound interfaces/applications are trivial or near critical. By examining Table 2, the results confirm the hypothesis H1 that control plane vulnerabilities are more severe than others. However, adding "Reputation" to DREAD model for ONOS does not yield different result. Therefore, hypothesis H2 does not validated. Adding reputation to other systems/apps may have different result.

4 Conclusion and Future Work

This paper assess the vulnerabilities severity of ONOS. We have used DREAD model to rate attacks found by DELTA Security framework and previous study using STRIDE threat model. We have assessed vulnerabilities and found that control plane vulnerabilities are severe and need to be patched on priority. Moreover, we have enhanced DREAD model by using an extra parameter "Reputation" named as DREAD-R. We realized that adding the "Reputation" to DREAD model does not yield different result using on ONOS. We will continue the assessment of SDN controllers such as ONOS. This can help the developers and user of ONOS to understand the threats and attacks. Mitigation techniques for vulnerabilities are recommended as future work.

Acknowledgment. This work was partly supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MIST) [2015-0-00533, Development of TII (Trusted Information Infrastructure) S/W Framework for Realizing Trustworthy IoT Eco-system], and partly supported by BK 21 plus program.

References

1. Nunes, B.A.A., Mendonca, M., Nguyen, X.N., Obraczka, K., Turletti, T.: A survey of software-defined networking: past, present, and future of programmable networks. IEEE Commun. Surv. Tutor. **16**(3), 1617–1634 (2014)
2. Kreutz, D., Ramos, F.M., Verissimo, P.E., Rothenberg, C.E., Azodolmolky, S., Uhlig, S.: Software-defined networking: a comprehensive survey. Proc. IEEE **103**(1), 14–76 (2015)
3. Shin, M.-K., Nam, K.-H., Kim, H.-J.: Software-defined networking (SDN): a reference architecture and open APIs. In: 2012 International Conference on ICT Convergence (ICTC), pp. 360–361. IEEE (2012)
4. SDN Architecture. Technical report, Open Networking Foundation (2014)
5. Arbettu, R.K., Khondoker, R., Bayarou, K., Weber, F.: Security analysis of OpenDaylight, ONOS, Rosemary and Ryu SDN controllers. In: 2016 17th International Telecommunications Network Strategy and Planning Symposium (Networks), pp. 37–44. IEEE (2016)

6. Yan, Q., Yu, F.R., Gong, Q., Li, J.: Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: a survey, some research issues, and challenges. IEEE Commun. Surv. Tutor. **18**(1), 602–622 (2016)
7. Lee, S., Yoon, C., Lee, C., Shin, S., Yegneswaran, V., Porras, P.: Delta: a security assessment framework for software-defined networks. In: Proceedings of NDSS, vol. 17 (2017)
8. Hong, S., Xu, L., Wang, H., Gu, G.: Poisoning network visibility in software-defined networks: new attacks and countermeasures. In: NDSS (2015)
9. Benton, K., Camp, L.J., Small, C.: Openflow vulnerability assessment. In: Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, pp. 151–152. ACM (2013)
10. Berde, P., Gerola, M., Hart, J., Higuchi, Y., Kobayashi, M., Koide, T., Lantz, B., O'Connor, B., Radoslavov, P., Snow, W., et al.: ONOS: towards an open, distributed SDN OS. In: Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, pp. 1–6. ACM (2014)
11. Meier, J.D., Mackman, A., Dunner, M., Vasireddy, S., Escamilla, R., Murukan, A.: Improving Web Application Security: Threats and Countermeasures. Microsoft Corporation (2003)
12. Adam Shostack: Threat Modeling: Designing for Security. Wiley (2014)
13. Thompson, D.R., Di, J., Sunkara, H., Thompson, C.: Categorizing RFID privacy threats with stride. In: Proceedings ACMs Symposium on Usable Privacy and Security held at CMU (2006)
14. Saitta, P., Larcom, B., Eddington, M.: Trike v. 1 methodology document [draft] (2005). http://dymaxion.org/trike/Trike_v1_Methodology_Documentdraft.pdf
15. Jürjens, J.: UMLsec: extending UML for secure systems development. In: Jézéquel, J.-M., Hussmann, H., Cook, S. (eds.) UML 2002. LNCS, vol. 2460, pp. 412–425. Springer, Heidelberg (2002). doi:10.1007/3-540-45800-X_32
16. Gilliam, D.P., Powell, J.D.: Integrating a flexible modeling framework (FMF) with the network security assessment instrument to reduce software security risk. In: Proceedings of Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, WET ICE 2002, pp. 153–158. IEEE (2002)
17. UcedaVelez, T., Morana, M.M.: Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis. Wiley, Hoboken (2015)
18. Schehlmann, L., Abt, S., Baier, H.: Blessing or curse? Revisiting security aspects of software-defined networking. In: 2014 10th International Conference on Network and Service Management (CNSM), pp. 382–387. IEEE (2014)
19. Chen, M., Qian, Y., Mao, S., Tang, W., Yang, X.: Software-defined mobile networks security. Mob. Netw. Appl. **21**(5), 729–743 (2016)
20. Shin, S., Yegneswaran, V., Porras, P., Gu, G.: Avant-guard: scalable and vigilant switch flow management in software-defined networks. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, pp. 413–424. ACM (2013)
21. FIRST. Common Vulnerability Scoring System v3.0: Specification Document
22. Selcuk, A.A., Uzun, E., Pariente, M.R.: A reputation-based trust management system for P2P networks. In: IEEE International Symposium on Cluster Computing and the Grid, CCGrid 2004, pp. 251–258. IEEE (2004)
23. Anantvalee, T., Wu, J.: Reputation-based system for encouraging the cooperation of nodes in mobile ad hoc networks. In: IEEE International Conference on Communications, ICC 2007, pp. 3383–3388. IEEE (2007)