Generalized Stochastic Petri Net Model Based Security Risk Assessment of Software Defined Networks

Laila M. Almutairi
Department of Electrical and Computer Engineering
Tennessee State University
Nashville, TN
lalmutai@my.tnstate.edu

Sachin Shetty
Virginia Modeling, Analysis and Simulation Center
Old Dominion University
Norfolk, VA
sshetty@odu.edu

Abstract— **Software-defined networking (SDN) is a networking paradigm to provide automated network management at run time through network orchestration and virtualization. A central controller realizes the automatic network configuration in SDN at run time by conforming to a control plane protocol (e.g., OpenFlow) and switches act as simple forwarding devices. However, SDN are susceptible to cyber attacks and there is a need to understand and quantify the cyber risks. In this paper, we present a model to analyze attacks on SDN and generate risk assessment scores that can aid mitigation. We build and analyze a Generalized Stochastic Petri Net (GSPN) model for Denial of Service attack in SDN using the PIPE tool. The results show all possible attacker paths during the attack. Moreover, they indicate that there is a direct relation between the risk score of the transitions and the average time the attacker needs to successfully perform individual attack action. These results can be used to improve countermeasures of SDN attacks in future work.**

Keywords— Software-defined networking (SDN); Security; Attack tree; Petri Net (PN); Generalized Stochastic Petri Net (GSPN); link layer discovery protocol (LLDP); Datapath ID (DPID); Denial of Service (DoS).

I. INTRODUCTION

Software-defined networking (SDN) is a network architecture that physically separates the network control from the network-forwarding infrastructure [1, 2]. In SDN, the software-based controllers provide a global view of the network, allowing the administrators to configure these resources and adjust the network traffic flow. This allows the underlying infrastructure to be abstracted for applications and network services, treating the network as logical entity. Control plan protocols (e.g., OpenFlow) are used to facilitate the interaction between the control plane and the network devices [3].

Industry and academia have proven that separating the control plane from the forwarding plane has many advantages. However, SDN introduces new threats and risk of attack that were not previously an issue with traditional network architectures [4]. Thus, understanding the security limitations of SDN is crucial to protecting the network and data. Most research work in this area has focused on building secure SDN structures by suggesting various solutions and techniques. Through modeling the network attacks, researchers can investigate and analyze the target network in more details. SDN studies have applied different modeling techniques to analyze SDN structures and protocols. Attack tree and STRIDE approaches are used to analyze OpenFlow protocol vulnerabilities [5, 6]. Yao et al. proposed a model by which they could analyze the security mechanisms of Forwarding and Control planes Separation Network Structure (FCSNS) in SDN [7]. The authors applied attack tree and Petri Net to represent the network structure and state transferring.

The current security risk models for SDN are not focused on quantifying the likelihood of a specific attack. In [5, 6] the authors used the attack trees to model OpenFlow protocol vulnerabilities, while in [7] the authors applied Petri Net and attack tree to model the structure of the SDN forwarding and control components. However, security administrators are often interested in evaluating the risk of their networks especially when these networks introduce new characteristics such as dynamic configuration and new protocols. Therefore, we use Generalized Stochastic Petri Net (GSPN) to model the SDN attacks and assess the network risks. GSPN attack model considers the dynamic behavior of SDN attacks such as sequential execution, concurrency and priorities of attackers' actions, synchronization of attacks' resources and states. Furthermore, The GSPN model allows users to assign weight to immediate transitions which facilitate the probabilistic simulation of the system.

This paper is organized as follows: Section II introduces attack trees, Stochastic Petri Net and Generalized Stochastic Petri Net; Section III presents an overview of the attacks that can take place in a software-defined network; and section IV demonstrate the GSPN attack model with an example. Finally, Section V and VI present evaluation of GSPN attack mode and conclusion and the recommendations for future work.

II. BACKGROUND

A. Attack Trees

Attack Trees (a term that was introduced by Bruce

978-1-5386-0595-0/17/$31.00 ©2017 IEEE

Schneier in [11]) are used to model system and network threats. In an attack tree, the root node represents the attacker's main goal, while the leaf nodes represent the methods by which that goal can be attained. Each node becomes a subgoal, and children of that node are steps towards the achievement of the subgoal [12]. The basic logical conditions of attack tree nodes are "OR" and "AND." OR nodes are used to represent alternatives steps, while AND nodes are used to represent different steps toward achieving the same goal [12]. Koli et al. reported OpenFlow protocol analysis and modeling using the STRIDE threat analysis methodology [13] and the attack tree modeling method [5, 6]. Fig. 1 shows an example of an OpenFlow attack tree as presented in [6]. It shows the required steps to learn the controller's behavior and achieve an information disclosure attack against an OpenFlow controller. Yao et al. represented security analysis and modeling of a Forwarding and Control planes Separation Network Structure (FCSNS) in an SDN. The authors used the attack tree and Petri Net to represent network structure and state transferring [7].

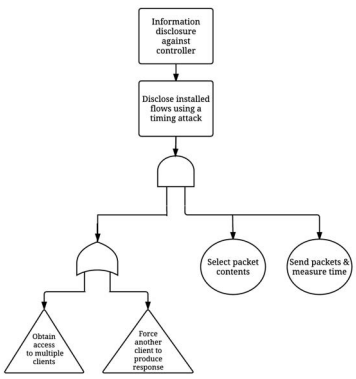

Fig. 1. Attack tree example [6].

B. Stochastic Petri Net and Generalized Stochastic Petri Net

Petri Net (or Place/Transition net) was designed by Carl Adam Petri in 1962 [8, 9] to describe distributed systems. It is directed, weighted and bipartite graphs that have two types of nodes: places and transitions [9]. Directed arcs connect places to transitions, or vice versa. The input arcs connect places to the transitions, while the output arcs connect transitions to places. Petri Net allows multiple arcs from one node to another. Places may carry tokens that mark their states. Arcs have weights that represent the number of tokens. Fig. 2 shows a simple Petri Net with two places and one transition. It also shows one token, which marks the state of the P0 place.

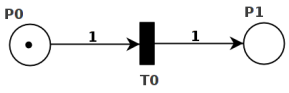

Fig. 2. Example of Petri Net model.

There are a number of extensions developed to expand the applicability of Petri Net. Stochastic Petri Net (SPN) is an extension of Petri Net where each transition is associated with a random firing delay whose probability density function is a negative exponential with rate λ [9, 10]. The dynamic behavior of a SPN can be described through a stochastic process that is a continuous time Markov chain (CTMC), where the wait times in states are exponentially distributed random variables. Generalized Stochastic Petri Net (GSPN) is an extension of SPN. The transitions in GSPN can be divided into two subtypes, immediate transitions and timed transitions. Also, the inhibitor arcs exist in GSPN [10].

Definition 1 A GSPN is a 6-tuple $(P, T, F, W, M_0, \lambda)$ where:

1. $P = \{p_1, p_2, ..., p_m\}$ is a finite set of places, $n \geq 0$.
2. $T = T_1 \cap T_2$, $T_1 = \{t_1, t_2, ..., t_m\}$ is a finite set of timed transitions, and each of these transitions is associated with a random delay time between enabling and firing; and $T_2 = \{t_{m+1}, t_{m+2}, ..., t_n\}$ is a finite set of immediate transitions, which can be fired randomly, and the delay is zero.
3. $F \subseteq (P \times T) \cap (T \times P)$ is a set of arcs; also there exist inhibitor arcs that can only form places to transitions and make the enable conditions to be disenabled.
4. W is a weight function of arcs: $F \rightarrow \{1, 2, 3, ...\}$.
5. $M_0 : P \rightarrow \{0, 1, 2, 3, ...\}$ is initial marking, where $(P \times T) = \varnothing \cap (T \times P) = \varnothing$.
6. $\lambda = \{\lambda_1, \lambda_2, ..., \lambda_n\}$ is a set of the firing rates corresponding to the timed transitions. Each rate is the average firing times of transition in unit time.

III. ATTACKS ON SDN

In this section, we present the threats and risk of attacks that are inherent in an SDN framework. The SDN framework has three layers: application, control, and data planes and interfaces between these layers as shown in Fig. 3.

A. Application Layer

The application layer is where the set of applications implement network control and operation logic through the application-control interface API. These applications facilitate network virtualization, network monitoring, intrusion detection (IDS), and load balancing [1]. Attacks on the application layer occur when unauthenticated applications are employed to access the network and reprogram it from a single location [14]. In addition, malicious applications could be used to insert flow rules into the network.

B. Control Layer

The control layer is the point which the controller or group of controllers determine the behavior of the network. The controller programs the network devices through the control-data interface API. Threats and attacks related to the control layer include unauthorized access, lack of authentication technique, hijacking, and denial of service (DoS) [14, 15]. Malicious applications could compromise an entire network. The lack of TLS/SSL adoption by most controllers could compromise the controller network device link, facilitating DoS and hijacking attacks [14, 15].

C. Data Layer

The data layer involves network devices such as routers

and switches. Some of the threats and attacks that are associated with the data layer are DoS, data modification, and data leakage. An attacker can use faulty devices to generate a DoS attack against SDN switches through flow table flooding. The lack of TLS/SSL could allow a hacker to introduce fraudulent rules into the switch flow tables and to subsequently modify or record sensitive data [14].

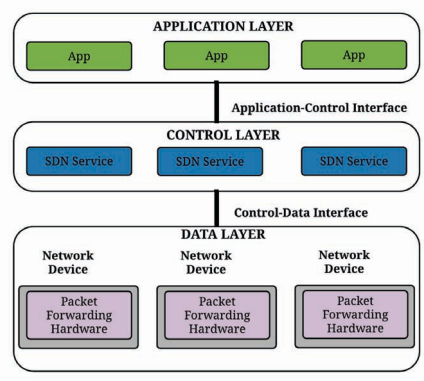

Fig. 3. SDN architecture [1].

IV. GSPN FOR SDN ATTACK MODELING

Attack modeling extracts critical network system information that can be subsequently used to perform comprehensive security analysis. We chose GSPN because of its the ability of providing the dynamic behavior of the system (attacks scenario in this work). The characteristics of the attacks in SDN are sequential execution, concurrency and priorities of attackers' actions, synchronization of attacks' resources and states. These properties of dynamic SDN attacks can be clearly shown using GSPN attack model. Furthermore, The GSPN model allows user to assign weight to immediate transitions that facilitate the probabilistic simulation of the system. In this section, we firstly describe the controller vulnerability that is used by the attacker to launch a DoS attack. Next, we examine this attack scenario using the GSPN model.

The SDN controller employs the topology management service to maintain the topology information of the network [16]. Internal link discovery is one SDN topology management service through which the controller uses link layer discovery protocol (LLDP) packets to detect internal links between switches [16]. An overview of the link discovery procedure is shown in Fig. 4. The controller encapsulates the LLDP packet, which contains Datapath ID (DPID) and the output port of switch S1, in a Packet-Out message and that is transmitted to switch S1. Switch S1 forwards it to all other ports in a broadcast manner. When the neighbor switch S2 receives the message from a port other than the connected controller port, S2 encapsulates the LLDP packet in a Packet-In message to the controller, which contains with the Port ID and DPID of switch S2. The controller can detect links between switches S1 and S2 and update the network topology. The same procedure is performed from switch S2 to S1 [16, 17].

The link discovery procedure has two security flows that the attacker can use to initiate a DoS attack. First, the controller is unable to confirm the integrity/origin of the LLDP packets during the link discovery procedure. Second, the transversed path of the LLDP packets can only contain OpenFlow-enabled switches [16, 17]. An attacker can use these issues to fabricate a fake inter-connected link between the SDN switches by modifying the LLDP packets or the LLDP relay. Later, the attacker can execute a DoS attack on the network.

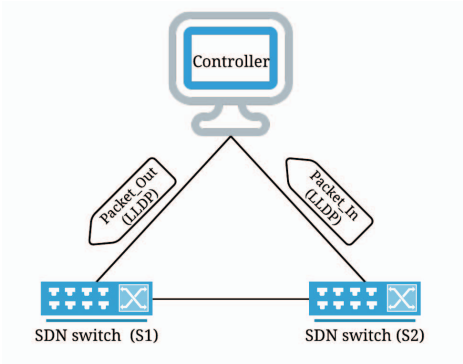

Fig. 4. The LLDP process in SDN environment [16].

For attack modeling, places represent security states and conditions, while transitions represent the attacker's actions. The input places are the preconditions for a transition (attacker action), and the output places are the outcomes of that transition. Tokens in places represent the concurrent attackers in the system and the progress of the attack.

Constructing a GSPN model for SDN attack scenarios involves number of steps. First, defining the main and the sub goals of the attacker. Second, enumerating all the possible security states of the SDN entities such as network LLDP packet, DPID of two switches, and switch with the smallest DPID. Identifying all the potential attacker actions that could affect changes in these security states. Finally, building the model up until reach the main goal. Building a GSPN model needs understanding the relationships between the security states and attacker's actions. Fig. 5 and Fig. 6 represent examples of the Boolean AND relationship between the security states. This relationship is used when one token in precondition places (attack route 1) is required to enable and then fire the transition (Get LLDP packet) and then the token will transfer to the next place (Original LLDP packet) to enable and fire the next transition (Modify LLDP packet) and reach the final place (Modified LLDP packet). AND with two tokens is used when there are two precondition places (Link fabrication attack, Switch with lower DPID) with at least one token in each of them are required to enable and then fire the transition (Link with the target switch) and reach the final place (DoS attack). Example of Boolean OR relationship is shown in Fig. 7. OR relationship is used to represent a scenario in which one of precondition places with at least one token (Modified LLDP packet, Network LLDP packet) is required to enable and then fire the associate transition and reach the final place (Link Fabrication attack). In this paper, we do not use an inhibitor arc, but we show how enable a transition that has an inhibitor arc in Fig. 8. The place that provides the inhibitor arc must not have a token [10].

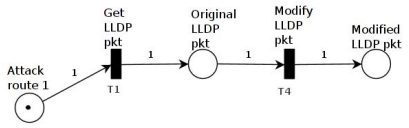

Fig. 5. Boolean AND relationship needs one token.

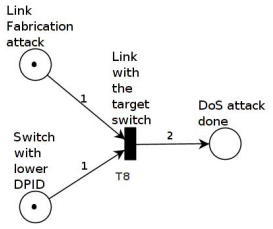

Fig. 6. Boolean AND with two tokens.

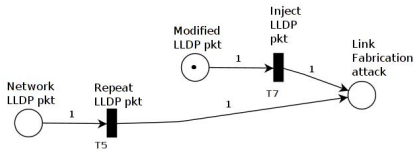

Fig. 7. Boolean OR relationship using Petri Net.

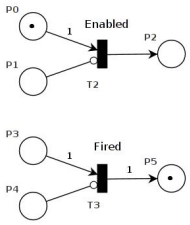

Fig. 8. Enabling the GSPN transition with inhibitor arc [9].

The GSPN attack model presented in Fig. 9 represents the complete DoS attack in an SDN. The places represent security conditions and the final states the attacker reaches. The transitions represent the actions the attacker is required to take to reach the next places. Table I outlines the corresponding places/transactions of the GSPN model. P0, P1, P2 are the initial states where the attacker can start his attack in SDN. P1 is a point for the attacker to initiate the ink fabrication attack either by fake LLDP injection, or by LLDP relaying. To achieve the overall goal P9, the attacker needs to reach the sub-goal states P7 and P8 where he can link the switch with lower DPID in the network with the target switch. T9 represents state of restarting the attack that required for model simulation.

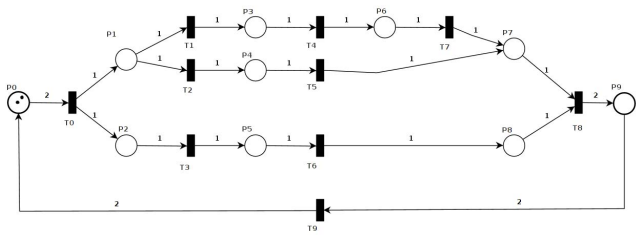

Fig. 9. Generalized Stochastic Petri Net model for DoS attack scenario in SDN using PIPE [18].

TABLE I. TRANSITIONS AND PLACES IN PETRI NET MODEL (FIG. 9)

Place	Description
P0	DoS attack start
P1	Attack route 1
P2	Attack route 2
P3	Genuine LLDP packet
P4	Network LLDP packet
P5	DPID of two switches
P6	Modified LLDP packet
P7	Link Fabrication attack
P8	Switch with lower DPID
P9	DOS attack done
Transition	**Description**
T0	Launch DoS attack
T1	Get genuine LLDP packet from external source
T2	Get LLDP packet from one target switch within the network
T3	Listen to LLDP packets
T4	Modify LLDP packet
T5	Repeat LLDP packet
T6	Find the lower DPID
T7	Inject LLDP packet
T8	Link with the target switch
T9	Restart DoS attack

V. EVALUATION

We evaluated the GSPN attack model by considering three attributes of the transitions: attack cost c_A, technical difficulty d_A and the discovering difficulty s_A. Table II shows the grade level standards [19]. The assignment of each transition's attribute requires knowledge of a specific system. We assigned values to the transitions according to the following rules:

- An attacker could launch an attack on any nodes of the system. Attacking on higher level node is more difficult for the attacker than lower level nodes. Also, the cost of the attack will be higher.
- An attacker should consider pre-states of the transition to launch the attack. Number and difficulty of required states causes more difficulty and cost for the attacker.

After value assignment for transitions, these three attributes were transferred into attackers' utility value P_A using the multi-attribute utility theory [20]. The following formula was applied to calculate the utility of each transition:

$$P_A = w_1.u_1(c_A) + w_2.u_2(d_A) + w_3.u_3(s_A) \quad (1)$$

$u_1(c_A), u_2(d_A), u_3(s_A)$ are the utility functions of the transition attributes; and the range of their values is between 0 and 1. w_1, w_2, w_3 are the weights of the attributes where $w_1 + w_2 + w_3 = 1$. To calculate the overall utility of each transition in the model, we set $w_1 + w_2 + w_3 = 1/3$ and utility functions $u_1(c_A) = u_2(d_A) = u_3(s_A) = u(c_x) = c/x$ and c = 0.2 to ensure the overall value occur within [0, 1] [19]. The overall occurrence probability P_A of each transition in the model is shown in Table III.

TABLE II. GRADE LEVEL STANDARDS [19]

Attack Cost (10^3) c_A	Grade	Technical difficulty d_A	Grade	Discovering Difficulty s_A	Grade
>10	5	Very difficult	5	Very difficult	1
6-10	4	Difficult	4	Difficult	2
3-6	3	Mediate	3	Mediate	3
0.5-3	2	Simple	2	Simple	4
<0.5	1	Very simple	1	Very simple	5

TABLE III. ATTRIBUTE VALUES OF MODEL TRANSITIONS

Transition	Attribute			Occurrence probability
	Attack cost c_A	Technical difficulty d_A	Probability to be discovered s_A	
T1	2	1	5	0.113
T2	3	2	3	0.078
T3	1	2	4	0.117
T4	3	2	3	0.078
T5	2	2	2	0.10
T6	1	1	5	0.147
T7	3	4	2	0.072
T8	4	4	2	0.067

We used PIPE (Platform-Independent Petri Net Editor) [18] to model and analyze the GSPN attack model of SDN. PIPE is an open-source tool that supports creating and analyzing Petri Nets. It has easy-to-use graphical user interface that allows a user to create standard Petri Net and Stochastic Petri Net models. It also allows a user to animate the model with random firing of transitions or interactive user manipulations. The analysis environment in this tool includes different modules such as steady state analysis, steady space analysis, and GSPN analysis [18].

First, we implemented the DoS model in PIPE as shown in Fig. 9. Next, we assigned weight to each of transitions as shown in Table III. We set w = 1 for T0 and T9. Since there is no timed transition in this case, we only obtained reachability graph, steady state analysis and average number of tokens per place. Fig. 10 illustrates the reachability graph of the model that can be used to explain the model behavior. Each of the graph nodes represent state of where the attacker located in the model states during the attack. The attacker starts with node S0 where its marking = {2, 0, 0, 0, 0, 0, 0, 0, 0, 0}, 2 is number of token in this graph node. Also, the reachability graph shows the steady state of this model that is deadlock free and bounded.

The designed GSPN model of the DoS attack was simulated fifty times using different number of initial random firings: 100, 300, 500, 700, 1000 and 1200. The variation of the token distribution with the same number of initial random firings is record. Table IV represented an example of simulation results. We used the average of each of the initial random firings for the final analysis as shown in Fig. 11.

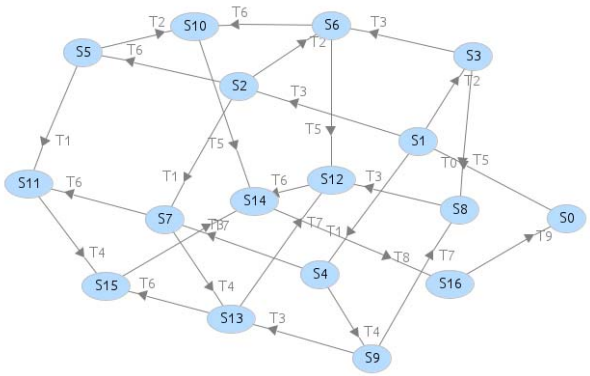

Fig. 10. Reachability graph of DoS attack model.

TABLE IV. SIMULATION RESULT OF THE MODEL WITH 100 TRANSITION FIRINGS AND 50 RUNS, USING PIPE [18]

Place	Average number of tokens	95% confidence interval (+/-)
P0	0.25743	0.01491
P1	0.17822	0.05658
P2	0.25743	0.06225
P3	0.18812	0.07556
P4	0.0495	0.07679
P5	0.20792	0.05087
P6	0.16832	0.05856
P7	0.16832	0.0494
P8	0.28713	0.06016
P9	0.23762	0.01053

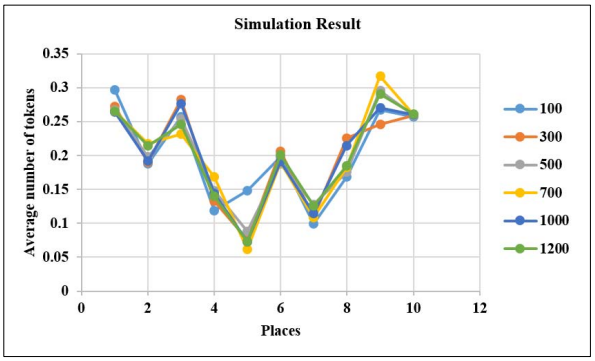

Fig. 11. Token distribution for 100, 300, 500, 700, 1000 and 1200 initial random firings.

The graph in Fig. 11 shows that each line reaches the highest point at P8 as this place represents the result of finding the switch with lower DPID (T6), which has the highest occurrence probability (0.147). P4 and P6 have the lowest points where they represent the outcomes of T2 and T4. These transitions have low probability (0.078) as shown in Table III. This means there is a direct relation between the transition probability and the average number of tokens of the state, which is the average time the attacker needs to successfully transfer from one state to another or to perform individual attack action. P7 is the output place of two transitions (T5: Repeat LLDP packet and T7: Inject LLDP packet) that have OR relationship. The result represents T_x+T_y that in this case equals to 0.172. In general, the model evaluation results help

to understand the attacker behavior during the attack and they can assist security analysis in future and develop corresponding countermeasures.

VI. CONCLUSION AND FUTURE WORK

In this paper, a novel Generalized Stochastic Petri Net attack modeling for SDN attack scenarios is introduced to facilitate an understanding of attackers' behaviors and method of analyzing the attacks. GSPN provides more flexibility and expressiveness than traditional attack trees. GSPN modeling is useful for qualitative and quantitative analysis. In this paper, the attack model is presented as a first step to understand the risk of potential SDN attacks and the dynamic behavior of the attacker. We have analyzed the threats to the controller protocol (LLDP) in SDN and built a GSPN attack model for one of its critical attack. Risk scores are assigned to transitions and multi-attribute utility theory is adopted to calculate the attacker's actions risk. Then, we have examined the dynamic of the attack scenario using PIPE. We have obtained reachability graph that indicates the model is deadlock free and bounded. Additionally, the simulation results show that there is a direct relation between the risk probability of the transitions and the average time the attacker needs to successfully perform individual attack action. These results can be used to improve the security assessment methods and develop countermeasures of SDN attacks in future work.

ACKNOWLEDGMENT

This work was supported by Office of the Assistant Secretary of Defense for Research and Engineering (OASD (R& E)) agreement FA8750-15-2-0120. The author, Laila Almutairi, was supported by Majmaah University through Saudi Arabian Cultural Mission, USA.

REFERENCES

[1] "Software-Defined Networking (SDN) Definition", *Open Networking Foundation*, 2016. [Online]. Available: https://www.open-networking.org/sdn-resources/sdn-definition. [Accessed: 24- Aug- 2016].

[2] S. Sezer, S. Scott-Hayward, P. Chouhan, B. Fraser, D. Lake, J. Foinnegan, N. Viljoen, M. Miller, and N. Rao, "Are we ready for SDN? implementation challenges for software-defined networks," IEEE Communications Magazine, vol. 51, no. 7, pp. 36–43, Jul. 2013.

[3] N. McKeown et al., "OpenFlow: enabling innovation in campus networks," SIGCOMM CCR, Mar. 2008.

[4] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A survey of security in software defined networks," IEEE Commun. Surveys & Tutorials, in print, 2015.

[5] R. Kloti, "OpenFlow: A security analysis," M.S. thesis, Inf. Technol. Elect. Eng. Depart., ETH Zurich, Zurich, Switzerland, Apr. 2013, MA-2012-20.

[6] R. Kloti, V. Kotronis, and P. Smith, "OpenFlow: A Security Analysis," in Proceedings of the 8th Workshop on Secure Network Protocols (NPSec), part of IEEE ICNP, ser. NPSec '13. IEEE, October 2013.

[7] L. Yao, P. Dong, T. Zheng, H. Zhang, X. Du and M. Guizani, "Network security analyzing and modeling based on Petri net and Attack tree for SDN", in International Conference on: Computing, Networking and Communications (ICNC), 2016.

[8] C.A. Petri. Kommunikation mit Automaten. PhD thesis, Technische University Darmstadt, 1962. Published by: Institut fur InstrumentellMathematik, Schriften des IIM Nr. 2 (in German), 1962. Also in: New York: Griffiss Air Force Base, Technical Report RADC-TR- 65{377, Vol. 1 (English translation), 1966.

[9] J.L. Peterson, Petri Nets, Theory and the Modeling of Systems. Upper Saddle River, N.J.: Prentice Hall, 1981.

[10] T. Murata, "Petri nets: Properties, Analysis and Applications," Proceedings of the IEEE, vol. 77, no. 4, pp. 541–580, 1989.

[11] B. Schneier, "Attack trees," Dr. Dobbs journal of Software Tools, vol. 24, no. 12, pp. 21– 29, 1999.

[12] V. Saini, Q. Duan, and V. Paruchuri, "Threat modeling using attack trees," Journal of Computing Sciences in Colleges, vol. 23, no. 4, pp. 124–131, 2008.

[13] S. Hernan et al., "Uncover Security Design Flaws Using The STRIDE Approach," http://msdn.microsoft.com/en-gb/magazine/cc163519.aspx, 2006.

[14] D. Kreutz, F. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," in Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. ACM, pp. 55–60, 2013.

[15] K. Benton, L. J. Camp, and C. Small, "OpenFlow vulnerability assessment," in Proc. 2nd ACM SIGCOMM Workshop HotSDN, pp. 151–152,2013.

[16] F. Pakzad, M. Portmann, W. L. Tan, and J. Indulska, "Efficient topology discovery in software defined networks," in IEEE ICSPCS, 2015.

[17] S. Hong, L. Xu, H. Wang, and G. Gu, "Poisoning network visibility in software-defined networks: New attacks and countermeasures," in NDSS'15, February 2015.

[18] J. Bloom, C. Clark, C. Clifford, A. Duncan, H. Khan, M. Papantoniou, T. Barnwell, M. Camacho, M. Cook, M. Gready, P. Kyme, and M. Tsouchlaris, "PIPE," Imperial College DoC Group Project, 2004.

[19] Dandan R., et.al., "A Novel Attack Tree Based Risk Assessment Approach for Location Privacy Preservation in the VANETs," in Proc. of the IEEE Intl. Conference on Communications, pp. 1–5, 2011.

[20] R. Sarin, Multi-attribute utility theory, Encyclopedia of Operations Research and Management Science 2001.