Network Security Analyzing and Modeling based on Petri net and Attack tree for SDN

Linyuan Yao[1], Ping Dong[*1], Tao Zheng[1], Hongke Zhang[1], Xiaojiang Du[2] and Mohsen Guizani[3]
[1]School of Electronic Information and Engineering, Beijing Jiao Tong University, P. R. China
[2]Department of Computer and Information Sciences, Temple University, USA
[3] Electrical and Computer Engineering Department, University of Idaho, ID, USA
Email: {11111020, pdong, zhengtao, hkzhang}@bjtu.edu.cn, xjdu@temple.edu, mguizani@ieee.org

***Abstract*—Due to the widespread research on Software Defined Networks (SDNs), its security has received much attention recently. But most of those attempts consider SDN security from the OpenFlow perspective. To the best of our knowledge, none so far has paid attention to the security analysis and modeling of Forwarding and Control planes Separation Network Structure (FCSNS) in SDN. Therefore, this paper provides a different approach to network security based on Petri net and Attack tree models. Our objective is to analyze the FCSNS security via the combination of model and state. This method represents the network structure and state transferring by way of Petri net. In addition, it introduces the security analysis method of STRIDE to build up the Attack tree model. Finally, we analyze FCSNS via the combination of Petri net and Attack tree model and present the results. Our results are very promising in using such models to achieve such security objectives.**

Keywords—Control; Forward; Separation; Petri net; Attack tree; OpenFlow; SDN

I. INTRODUCTION

With the rapidly increasing development of the Internet and the boom of network users, traditional IP Internet architectures are facing various inevitable problems. Despite their widespread adoption, IP networks are complex and hard to manage [1]. It is difficult to update the configuration of the network devices according to predefined policies to respond to delay, load, and changes. To make matters even more difficult, current networks are also vertically integrated: the control and data planes are bundled together [2]. Along with the aim to solve the long-term problems, the proposals are discussed heatedly, among which the separation of control plane and forwarding plane is widely popular. The most obvious phenomenon is Software Defined Network (SDN) [3][4], which brings growing concerns and higher expectations.

Fast expansion of the Internet moves personal privacy, confidential documents of the government, and account details of the financial institutes from offline to online. The security of the Internet has become the highest priority of current IT developments. OpenFlow, as a representative protocol of SDN, receives much attention in terms of security. Several security approaches based on OpenFlow have been provided. A new security application development framework, FRESCO, was introduced in [5]. It provided a Click-inspired programming framework for security researchers to implement, share, and compose together. In [6], OpenFlow Random Host Mutation (OFRHM) was presented. Because of IP mutation being transparent to end-hosts and virtual IP, OFRHM can defend against scanning-based attacks. In [7], Guang Yao *et al.* proposed an OpenFlow-based mechanism named VAVE, which can improve the SAVI solutions by solving source address validation problems.

Methods to detect DDoS using OpenFlow have also been proposed in recent years. In [8], a lightweight method based on traffic flow features was presented for detecting DDoS attacks. Suh *et al.* [9] proposed a content-oriented networking architecture, which reacted to resource exhaustive attacks like DDoS. Chu,YuHunag *et al.* [10] proposed a new research idea utilizing OpenFlow and LISP technologies. They implemented a DDoS defender on an OpenFlow-enabled switch to realize an autonomic self-defense concept.

There is some research in other aspects of OpenFlow security as well. Porras *et al.* introduced FortNOX, which provided role-based authorization and security constraint enforcement for the NOX OpenFlow controller in [11]. In [12], OpenSAFE was proposed to enable the arbitrary direction of traffic to monitor applications at line rates. In [13], the authors performed a security analysis of OpenFlow by using STRIDE [14] and Attack tree modeling. All of these research attempts consider the security of SDN only from the perspective of OpenFlow.

However and to the best of our knowledge, there is still little attention given to the security analysis of Forwarding and Control planes Separation Network Structure (FCSNS), as well as SDN. Reference [15] and [16] analyze the security of SDN architecture. Six vulnerabilities in SDN control platforms were listed in [15]. Seven main potential threat vectors and related possible solutions were described in [16]. Kreutz *et al.* discussed some mechanisms and techniques that can be used on the design of a secure and dependable control platform. These two articles only described possible problems and corresponding solutions on SDN architecture; they lacked further modeling analysis for connection between state and vulnerability.

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant No. 2014JBM004, 2015JBM001, in part by Beijing Higher Education Young Elite Teacher Project under Grant No. YETP0534, in part by the NSFC under Grant No. 61232017, in part by National Science and Technology Major Projects of the Ministry of Science and Technology of China No. 2013ZX03006002, in part by the Cooperation Projects by Production, Study and Research under Grant No. YB2014060041.
(Corresponding author: Ping Dong)

978-1-4673-8579-4/16/$31.00 ©2016 IEEE

Therefore, we provide an Approach of network security Analyzing and Modeling based on Petri net and Attack tree (AAMPA) to analyze the security of FCSNS via the combination of model and state. The major contributions are summarized as follows.

- Dividing the FCSNS structure into three parts as user access, data transmission, and control command distribution, and modeling in Petri net [17] for the three partitions to represent network structure and state transferring.
- Building up the Attack tree model [18] for the Place and Transition in Petri net by introducing the security analysis method of STRIDE.
- Analyzing the security of FCSNS via Petri net model and Attack tree model.

The rest of the paper is organized as follows: Section II gives a more detailed description of AAMPA. We present an explanation of AAMPA as SDN as an example in Section III. Finally, we conclude this paper in Section IV.

II. DESCRIPTION OF PROPOSED AAMPA

This section provides a detailed explanation of AAMPA. AAMPA includes a total of four parts from the analysis of the network architecture to security analysis.

A. Design of network topology and Data Flow Diagram (DFD)

The control and data planes are decoupled in FCSNS. First, we divide the network topology into two parts on the basis of control plane and forwarding plane. The control plane includes Administrator (A) and Controlling Device (CD). Terminal (T) and Forwarding Device (FD) are assigned to the forwarding plane. The topology is shown in Fig.1.

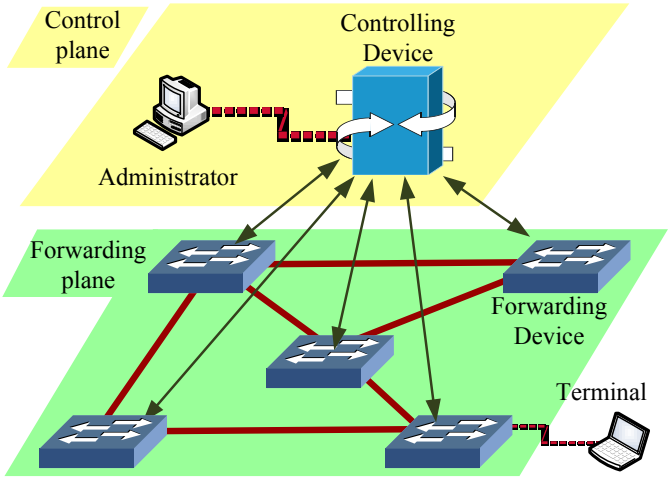

Fig. 1 Forwarding and Control planes Separation Network Structure

Next, we lay out the topology as the DFD [14]. Considering the scale and complexity of DFD, two FDs are appropriate for the analysis, which are each connected with one T in Fig.2. According to the features of DFD, Data Flow (DF), Transmission Channel (TC), and Trust Boundary (TB) are shown. DF1, 2, 3, 4 represent the DFs between A and CD, CD and FD, FD and FD, FD and T, respectively. There are two TBs, which are determined by three parts that access users, network, and administrator. TB1 is located between FD and T. TB2 is located between A and CD.

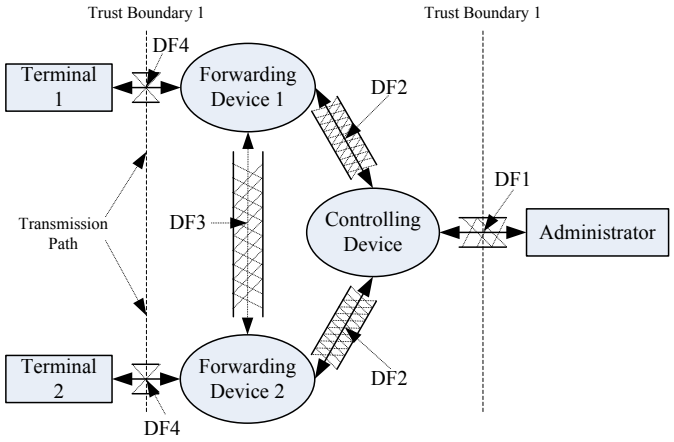

Fig. 2 Data Flow Diagram of FCSNS

B. Petri net modeling

There are three basic active network states in FCSNS, which are user access, data transmission, and control command distribution. According to the partitions of TBs and states in FCSNS, we regulate Fig.1 into three connection modes, such as CD-FD-T, FD-T, A-CD-FD. For a better display of the network states, it is necessary to model the modes by Petri net. The modes demonstrate the entities, links, and states graphically and logically. The details about Petri net modeling will be shown in section III.

C. Security analysis and Attack tree modeling

In this step, STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege) is quoted to assist in analyzing the vulnerability of elements in FCSNS. Depending on the vulnerability, Attack tree models are built in combination with Petri net models in step B.

D. Relation and Summary

We consider the Attack tree model and the Petri net model together to analyze the security of FCSNS. The purpose of this step is to make the connection between the attack source and the vulnerability source of elements from the perspective of the overall network structure, and provide the references for the security study.

III. DETAILS ABOUT AAMPA BASED ON SDN

As a typical example of FCSNS, SDN has received much attention in recent years. SDN is defined with four pillars as described in [2]. 1) The control and data planes are decoupled;

2) Forwarding decision is based on the flow, instead of the destination; 3) Control logic is removed from a traditional router to an external entity, SDN controller, or NOS; and 4) The software applications running on top of the SDN controller are able to program the network. In consideration of the above four pillars, SDN is chosen to be used to specify AAMPA in this section.

A. Design of network topology and DFD

Fig.1 shows the generalization of FCSNS. In actual discussion, we replace CD and FD with Controller (C) and Switch (S), respectively. The number of C and S is set to 1 and 2.

B. Petri net modeling

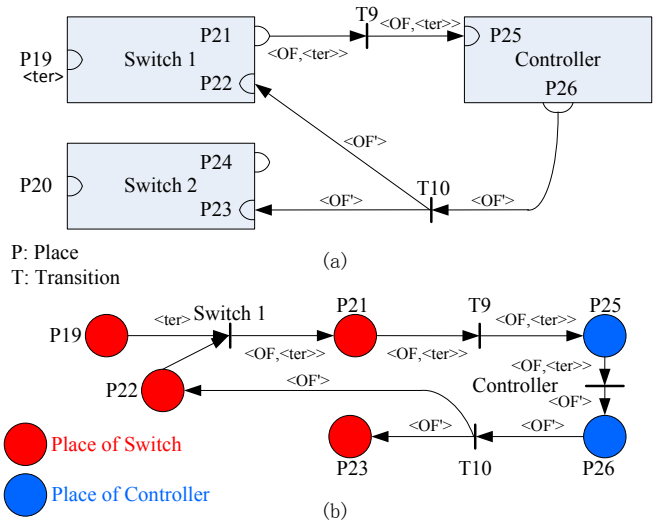

Fig. 3 User access

As described in step B of Section II, Fig.2 is divided into three connection modes, dependent on user access, data transmission, and control command distribution in Fig.3, Fig.4, and Fig.5. There are 2 diagrams within Fig.3, Fig.4, and Fig.5, which are represented by way of Petri net in order to make a more vivid state. Diagram (a) presents a top-level architecture composition, while diagram (b) is a more detailed state of (a). For further attack analysis in the next step, some specific abbreviations of the key aspects are given in the diagram. OpenFlow is the reference of the information as the instruction as shown in TABLE I. TABLE II shows the descriptions about Place and Transition which are two basic elements required in Petri net [17].

Fig.3 shows the Petri net model of user access. User access is the message-forwarding process of entities during a new Terminal accessing the network. If the Terminal is connected to the network by Switch 1 for the first time (<ter> sent from P19 to Switch 1), Switch 1 will report the network change (<OF,<ter>>) to the Controller (through T9). After receiving the message from Switch 1, the Controller will make a decision about certification for Terminal and send the control message (<OF'>) to Switch 1, 2 (through T10). This process includes 3 entities, Terminal, Switch and Controller.

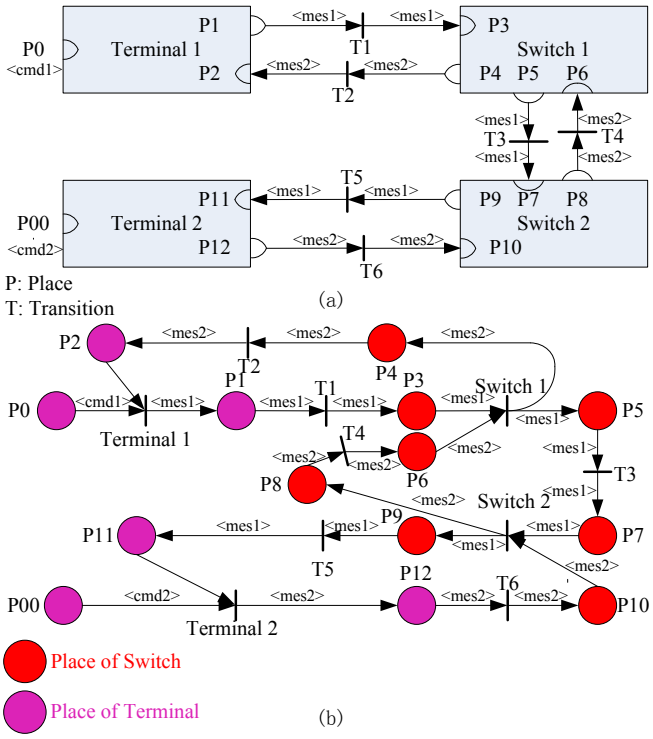

Fig. 4 Data transmission

The data transmission shows the state of data transmissions between two Terminals in Fig.4. The state is based on Terminal 1, 2 getting the certifications from the Controller successfully. When a user of Terminal 1 or 2 wants to get some information from Terminal 2 or 1, the user will send a command to Terminal 1 or 2 (<cmd1 or 2> by P0 or P00). Because Switch 1, 2 know how to forward the message from Terminal 1, 2 (<mes1/2>), Switch 1, 2 will receive and forward the message to the next Switch 2, 1 (through T3, 4) or Terminal 1, 2 (through T1, 2, 5, 6). The Terminal and Switch should exist in this part.

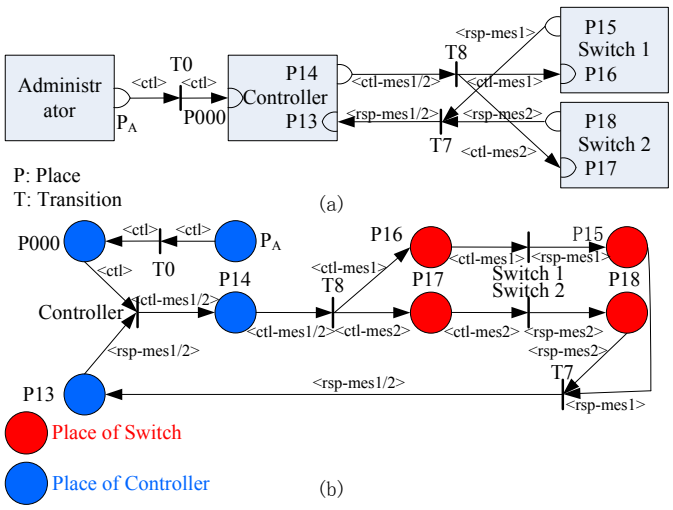

Fig. 5 Control command distribution

Control command distribution in Fig.5 is used to demonstrate that if Administrator operates Controller (<ctl> by P000), Controller will send the command to Switch or inform Switch to update the version of software, and so on (<ctl-mes1/2> through T8). The response (<rsp-mes1/2>) will be sent to Controller (through T7) after Switch 1, 2 process the message. The whole process is completed between Administrator, Controller and Switch, therefore there is no Terminal in Fig.5.

TABLE I. VARIABLES IN FIG.3,4,5

Variable	Description
ter	Terminal information
OF	OpenFlow head message
OF'	OpenFlow actions
cmd1/2	User command 1 and 2
mes1/2	Terminal 1 message and Terminal 2 message
ctl	Control command
ctl-mes1/2	Control message 1 and 2
rsp-mes1/2	Response message 1 and 2

TABLE II. LEGENDS FOR FIG. 3, 4, 5

Place	Description	Place	Description
P0	User 1 command message	P15	Response message 1
P00	User 2 command message	P16	Control message 1
P000	Administrator control message	P17	Control message 2
P1, P3, P9, P11, P5, P7	Terminal 1 message	P18	Response message 2
P2, P4, P10, P6, P8, P12	Terminal 2 message	P19	Terminal 1 access request
P13	Response message 1 and 2	P21, P25	Terminal 1 access request with OpenFlow head message
P14	Control message 1 and 2	P22, P23, P26	OpenFlow decision to Terminal 1 access request
Transition		**Transition**	
T1, T3, T5	Transmit <mes1> to Switch 1, Switch 2, Terminal 2	T8	Transmit <rsp-mes1/2> to Controller
T2, T4, T6	Transmit <mes2> to Switch 2, Switch 1, Terminal 1	T9	Transmit <OF, <ter>> to Controller
T7	Transmit <ctl-mes1> and 2 to Switch 1 and 2	T10	Transmit <OF'> to Switch 1 and 2

C. *Security analysis and Attack tree modeling*

In this step, STRIDE method is quoted to be combined with the three connection modes in the second step to create Attack tree model based on SDN. Before using STRIDE, we need to extract the four elements from the research system, namely data flow, entity, data storage and process, and then make an analysis for the six security threats centered on the elements. The security and process of data storage are mostly influenced by the design of the software itself, instead of network structure, in most systems. However, the vulnerabilities related to the architecture are the entity and data flow among all the factors. As far as the main purpose of FCSNS in this paper, the analysis will be oriented as the entity and data flow.

TABLE III lists the details of each element and the potential relative attacks.

TABLE III. ELEMENT AND ATTACK

Element	Description	Type of attack
Entity(4)	Administrator (A), Controller (C), Switch (S), Terminal (T)	Spoofing, Repudiation
Data Flow(4)	Data Flow between A and C (DF1), Data Flow between C and S (DF2), Data Flow between S and S (DF3), Data Flow between S and T (DF4)	Tampering, Information disclosure, Denial of service

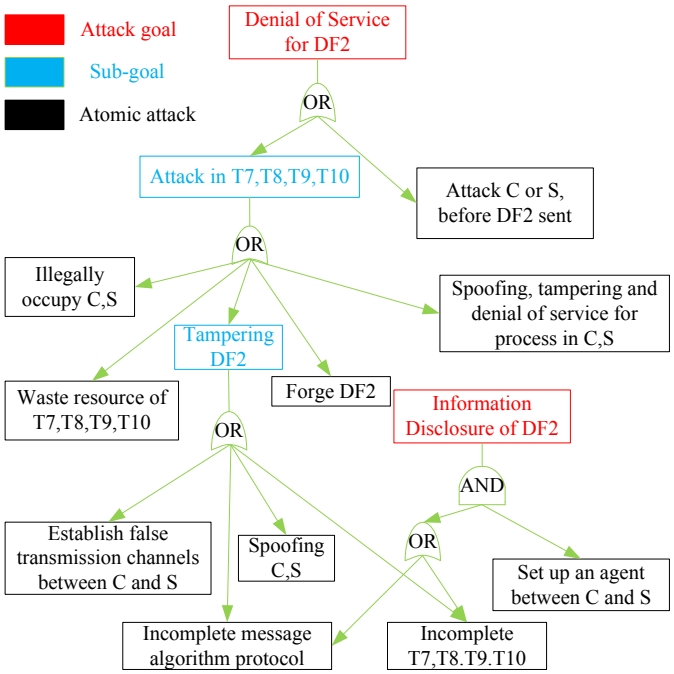

Fig. 6 An Attack tree for DF2

Attack tree model describes attacks towards any system as a logical function of atomic attacks. A successful security breach due to a cyber attack is called an attack goal. Successive subordinate security breaches are called attack sub-goals. The sub-goals can be broken down further to successive events called atomic attacks. The atomic attacks and attack sub-goals are connected to the attack goal using "AND/OR" nodes. Conventional Attack tree formulation uses "AND/OR" nodes to represent attack sub-goals. For lack of space, we represent the Attack tree modes for DF2 in Fig.6 and A in Fig.7, instead of each element described in TABLE III.

Fig.6 shows an Attack tree with the 'Denial of Service for DF2' and 'Information Disclosure of DF2' attack goals colored in red. As one of the attacks for DF2, Tampering attack, colored in green, is one of the sub-goals in this model. The other sub-goal is 'Attack in T7, T8, T9, T10'. From the figure, we can find ten kinds of atomic attacks colored in black. For DF2, there are three main attack types, Denial of service, Tampering, and Information disclosure. Among them, Denial of service is the most easily accomplished. Any of the nine atomic attacks will lead to Denial of service. Information disclosure is much harder than the other ones; the attackers will have to accomplish setting up an agent if they want to disclose the information.

The attack goal in Fig.7 is 'Repudiation for A'. Replay attack and Spoofing are the major attack sources. For Spoofing, we should pay more attention to transmission channels and devices, where the legal certification exists.

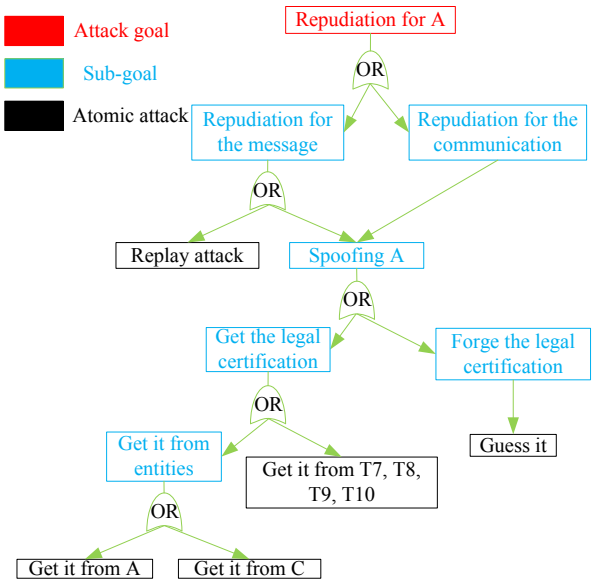

Fig. 7 An Attack tree for A

D. *Relation and Summary*

According to the analysis and modeling in steps B and C, we transform Fig.3, Fig.4, Fig.5, Fig.6 and Fig.7 into TABLE IV. We observe that the distribution and number of attack types for DF2 and A are given in details. Obviously, the security of Controller and Switch is significant for DF2. Accordingly, we should improve the integrity on the communication and transmission channel between Controller and Switch. For A, we should guard against agents.

TABLE IV. PETRI NET AND ATTACK STYLE

Attack object	Attack type	Number of type
S	Illegally occupy; Spoofing; Spoofing, tampering and denial of service for process in C,S	5
C	Illegally occupy; Spoofing; Spoofing, tampering and denial of service for process in C,S	5
A	Repudiation (A replay attack; Spoofing A (get the certification from A,C,P000; get the certification from T7, T8, T9, T10; Guess the certification))	4
T7,T8, T9,T10	Incomplete, waste resource, incomplete message algorithm protocol, establish false transmission channels	4
P000	Set up an agent	1
P13-P18,P21-P26	Attack before DF2 sent, Illegally occupy, forge,	3

IV. CONCLUSION

In this paper, we proposed a new security scheme for SDN called AAMPA to analyze the security of FCSNS via the combination of model and state. AAMPA divides the FCSNS structure into three parts as user access, data transmission, and control command distribution, and then builds up the models by way of Petri net and Attack tree to represent network structure and state transferring. Finally, we analyze the security of FCSNS using Petri net model and Attack tree model.

To clarify AAMPA, we chose a typical architecture as well as SDN to analyze the security of entities A and data flow. In our future research agenda on this topic, we plan to continue our research on other types of attacks according to the analysis given in this paper. We are confident that this work will contribute to FCSNS or SDN research for network architectures and standards.

REFERENCES

[1] T. Benson, A. Akella, and D. Maltz, ''Unraveling the complexity of network management,'' in Proc. 6th USENIX Symp. Networked Syst. Design Implement., 2009, pp. 335–348.

[2] D. Kreutz, F.M.V. Ramos, P.E. Verissimo, et al., "Software-defined networking: A comprehensive survey," proceedings of the IEEE, 2015, 103(1): 14-76.

[3] N. Mckeown, "How SDN will shape networking," Mar. 2015. Available: http://www.youtube.com/watch?v= c9-K5O_qYgA.

[4] S. Schenker, "The future of networking, the past of protocols," Mar. 2015. Available: http://www.youtube.com/ watch?v=YHeyuD89n1.

[5] S. Shin, P. Porras, V. Yegneswaran, et al., "FRESCO: Modular Composable Security Services for Software-Defined Networks," NDSS. 2013.

[6] J.H. Jafarian, F. Al-Shaer, Q. Duan, "Openflow random host mutation: transparent moving target defense using software defined networking," Proceedings of the first workshop on Hot topics in software defined networks. ACM, 2012: 127-132.

[7] G. Yao, J. Bi, P. Xiao, "Source address validation solution with OpenFlow/NOX architecture," Network Protocols (ICNP), 2011 19th IEEE International Conference on. IEEE, 2011: 7-12.

[8] R. Braga, E. Mota, A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," Local Computer Networks (LCN), 2010 IEEE 35th Conference on. IEEE, 2010: 408-415.

[9] Y. Choi, "Implementation of Content-oriented Networking Architecture (CONA): A Focus on DDoS Countermeasure," Proceedings of European NetFPGA developers workshop. 2010.

[10] Y.H. Chu, M.C. Tseng, Y.T. Chen, Y.C. Chou, Y.R. Chen, "A novel design for future on-demand service and security," 2010 IEEE 12th International Conference on Communication Technology. 2010: 385-388.

[11] P. Porras, S. Shin, V. Yegneswaran, et al., "A security enforcement kernel for OpenFlow networks," Proceedings of the first workshop on Hot topics in software defined networks. ACM, 2012: 121-126.

[12] J.R. Ballard, I. Rae, A. Akella, "Extensible and scalable network monitoring using opensafe," Proc. INM/WREN, 2010.

[13] R. Klöti, V. Kotronis, P. Smith, "Openflow: A security analysis," Proc. Wkshp on Secure Network Protocols (NPSec). IEEE, 2013.

[14] H. Shawn, L. Scott, O. Tomasz, S. Adam, "Uncover Security Design Flaws Using The STRIDE Approach," Mar. 2015. Available: http://msdn.microsoft.com/en-gb/magazine/cc163519.aspx.

[15] K. Benton, L.J. Camp, C. Small, "Openflow vulnerability assessment," Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. ACM, 2013: 151-152.

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

[17] L. James, Peterson, Petri net Theory and the Modeling of Systems, Englewood cliffs: Prentice-Hall, 1981:7-115.

[18] B. Schneier, Attack Trees, Dr. Dobb's Journal, vol. 24, no.12, pp.21–9, 1999.