Security Evaluation Methodology for Software Defined Network Solutions

Jean Claude Nikoue
Concordia University of Edmonton
Edmonton, Canada
jnikoue@student.concordia.ab.ca

Sergey Butakov; Yasir Malik
Concordia University of Edmonton
Edmonton, Canada
{sergey.butakov; yasir.malik}@concordia.ab.ca

***Abstract*— Software Defined Networking (SDN) has introduced both innovative opportunities and additional risks in the computer networking. Among disadvantages of SDNs one can mention their susceptibility to vulnerabilities associated with both virtualization and the traditional networking. Selecting a proper controller for an organization may not be a trivial task as there is a variety of SDN controllers on the market and each of them may come with its own pros and cons from the security point of view. This research proposes a comprehensive methodology for organizations to evaluate security-related features available in SDN controllers. The methodology can serve as a guideline in the decisions related to SDN choice. The proposed security assessment follows a structured approach to evaluate each layer of the SDN architecture and each metrics defined in presented research has been matched with the security controls defined in NIST 800-53. Through the tests on actual controllers the paper provides an example on how the proposed methodology can be used to evaluate existing SDN solutions.**

Keywords*— *Software-Defined Networking, SDN security, Security evaluation

I. INTRODUCTION

Software Defined Networking (SDN) is a networking paradigm in which the control and management of the network is separated from the traffic forwarding primitives [1], [2]. The management of the network is done at the Application plane, the traffic control is centralized at the Control plane and traffic forwarding is done at the Data plane.

Separating the Data and the Control planes provides an SDN network enough resilience to merge the advantages of system virtualization and cloud computing [3]. However, along with virtualization and the traditional networking issues, SDN has some major security vulnerabilities associated with the its Control plane. These vulnerabilities create a number of concerns for SDN adoption. With the increasing number of SDN solutions and its applications, providing a security evaluation methodology for SDNs would help in determining which SDN to implement in an organization that best fits the needs and security compliance requirements.

In this research, an SDN security evaluation methodology is proposed that can serve as a guideline for evaluating security implementations in an SDN infrastructure. The methodology looks at all aspects of the SDN architecture: Application, Control, and Data planes. Each evaluation metric has been defined from list of common vulnerabilities specific to SDNs infrastructure and system virtualization and is mapped from security controls defined in NIST 800-53 [6] and the Security Technical Implementation Guide (STIG) for SDNs. To validate the proposed methodology, four SDN controllers - OpenDayLight (ODL) [7], ONOS [8], Floodlight [9] and Ryu [10] have been deployed and tested in a simulation environment. Discussion on the result obtained from the simulations is provided to validate comprehensiveness of the methodology.

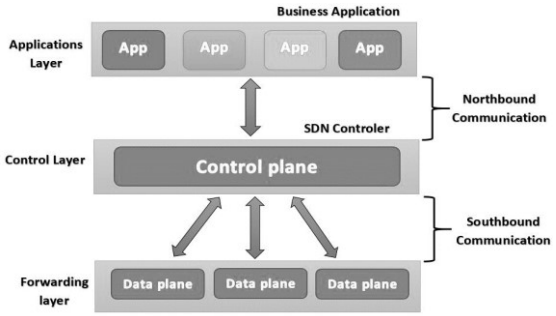

Fig. 1. SDN architecture

II. 2 RELATED WORKS

Security issues brought by SDNs do not just come from the new approach to net-working but have their roots in virtualization. Concerns raised by virtualization vary from performance to security. Those security concerns can be grouped as follows:

- Problems with Hypervisor: VM escape [11] [12]; Single point of failure [12] [13], improper input validation [14], failure to maintain processes within the bounds of memory buffer [14], [13]; improper authentication [14], [13], [15], and authentication bypass by capture-replay [14], [16].
- Problem with communication: DoS/DDoS attacks on the host machine [12], [17]; Unauthorized access to network due to inappropriate authorization [16], [14], [15].

Common vulnerabilities of an SDN can be linked to each of the three planes. An overall list of the vulnerabilities at each plane was identified in [18]. At the Data plane, security vulnerabilities can be associated to the protocols governing

978-1-7281-1288-6/19/$31.00 ©2019 IEEE

data transfer in the SDN. Historically, OpenFlow and OpFlex are the first two protocol used in SDN implementation to establish communication between the control plane and the Infrastructure layer of an SDN. Those two protocols have been known to run over Transport Layer Security (TLS) or TCP connections with low protection.

Table 1 outlines a list of vulnerabilities identified by previous researches. This list of vulnerabilities is structured following an SDN architecture. Review of previous works has demonstrated that most of the security recommendations are directed more towards SDN developers in the quest to build secure SDN and network architects who are deploying an existing SDN. On another hand. recommendations listed in STIG for general networking implementations do not look at problems specific to SDNs thus may not be really used to evaluate security of SDN implementations [24], [4]. Limited research has been done to assist IT management and network professionals in evaluating security of various SDN solutions. Such research is required to provide management with advice on the suitability of the SDN solutions for organization's needs from the security perspective.

TABLE I. SDN VULNERABILITIES, PROPOSED SOLUTIONS, AND LIMITATIONS

Layer	Vulnerabilities identified	Controls	Potential control limitations
Application Plane	Lack of authentication and authorization of applications [18]	Use of a Security Enforced Kernel (SEK) to authenticate applications and detect rule conflicts [25]	Limited to Floodlight controller and requires and administrator to pre-sign the applications java class
	Lack of access control and accountability [18]		
	Lack of application isolation can lead to inconsistent flow rules [8]	Check flow rules contradiction in real time and implement role-based authorization through a security enforcement kernel [26]	Requires complex algorithm to determine application security level.
Northbound Communication	Flow rules injections due to non-verification of the application by the controller [18]		
	Weak authentication between the applications and the controller, which may lead to spoofing [3]	Use of TLS: avoid eavesdropping, spoofing, and validate the identity of components [3]	TLS is optional and limited number of vendors supports it. [3]
Control Plane	DoS Attacks [3] [18]	Rerouting traffic to a middlebox [3] [20]	The solution it limits the flexibility and scalability of SDN [3]
	Implementation of rogue controller to hijack put the entire SDN [3]	Hardening and monitoring of the controller [3]	Depends on the implementation and on the underlying operating system [3]
	Threats from unauthorized access [18]		
	Controller hijacking or compromise [18]		
	Threats due to Scalability [18]	Use of Distributed controller [27] [29]	Can increase latency in the network
Southbound Communication	Eavesdropping and spoofing possible due to no encryption	Use of Security manager to encrypt communication between controller and switches [23]	Requires implementation of a PKI at the controller level and may increase latency
	Weak authentication can lead to Man-in-the-Middle attack at this level [3]	Use of a naming manager at the controller to name each device on the network and authenticate them before communication is established [23]	Limits flexibility and scalability because naming is done at the initial stage of the network before any communication is established.
	Unauthorized access to network due to inappropriate authorization [3]		
Data Plane	Flooding attacks [18]	Configure nodes to filter out illicit connections [22]	Requires additional configuration on the entire SDN network.

III. SECURITY EVALUATION METHODOLOGY

The evaluation methodology described in this section intends to help organizations to perform security evaluation of SDN solutions. The methodology is based on top down approach to SDN architecture in order to evaluate existence and implementation of the recommended security controls. Set of pre-selected controls based on NIST 800-53 document forms the basis for the evaluation. The proposed methodology should be used as follows:

- Step 1: Evaluation of controls. The user should evaluate existence and efficiency of the controls in a specific SDN implementation. As a result of this step the user will get a list of existing, non-existing and compensating controls that address known vulnerabilities in SDNs.
- Step 2: Scoring and analysis of SDN. On this step each of the controls will be assigned a score. The score should be based on the importance of each component of the SDN to the organization. This score must be based on the organization security policy and

standards. Analysis of results obtained from this step will pro-vide quantitative evaluation of the security posture of evaluated SDN controller in the context of specific organization.

After a review of well-known set of recommendations in the NIST 800-53 Rev. 5 document, a number controls have been selected based on their relevance and applicability in an SDN environment. For example, the first control defined in the methodology, from the Identification and Authentication family, IA-3 talks about device identification and authentication. This control specifies that each organization defined devices needs to be uniquely identified and authenticated before establishment of any kind of connection [6]. Based on the NIST 800-53 and analysis pro-vided in table 1 a few controls have been selected as per the following categories of control groups: Audit and Accountability (AU); Configuration Management (CM); Contingency Planning (CP); Identification and Authentication (IA); System and Communication Protection (SC); System and information Integrity (SI). The follow-ing subsections outline the controls that constitute the backbone of the proposed methodology in the SDN context.

A. Application Plane and Northbound Communication

The SDN controller receives configuration instructions and security flow rules from applications on the Northbound API.

- IA-3 Device identification and authentication: Each application needs to be uniquely authenticated prior to sending any instruction to the SDN controller.
- IA-7 Cryptographic module authentication: all communication need to be mutually authenticated by the Northbound APIs and SDN controller using Federal Information Processing Standard (FIPS) - approved message authentication code algorithm.
- IA-10 Adaptive authentication: an application manager needs to be implemented as a security enforced kernel to avoid more than one application performing the same role.
- SC-8 Transmission confidentiality and integrity: this layer of the SDN, communication between the Application plane and the Control plane must be encrypted to ensure confidentiality of data shared between the two layers.
- An alternative to the previous approach is based on SC-37 Out-of-band channels: create a dedicated network for all northbound traffic through an out-of-band channel.
- AU-2 Audit events, AU-10 Non-repudiation: each request or rules transferred between an application and the SDN controller must be kept in a log with enough details to trace the sender and the receiver. This will help check for accountability in case there is any misconfiguration or attack from the application plane.

B. Control Plane

The Control plane constitute of the SDN controller which receives rules from the Application plane to deploy, configure and manage the network. The SDN controller also receives information about new devices and traffic from the Data plane.

- CP-7 Alternate Processing site and SC-36 Distributed Processing and storage: a cluster of SDN controllers with load balancing capability is required.
- SA-2 Allocation of resources: The SDN controller must be deployed on a dedicated computer with enough computing resources to handle all traffic.
- SC-5 Denial of service protection: flow control application need to be deployed to detect possible DoS and DDoS attacks.
- SC-36 Distributed Processing and storage: computer that hosts the SDN controller must have at least two network interfaces with link aggregation.
- SI-4 System Monitoring: A Host Intrusion Detection System (HIDS) needs to be implemented on the machine hosting the SDN controller to monitor and report any system configuration changes and prevent anomalous activities.

C. Southbound communication

This layer covers all communication between the Data plane and the SDN controller, therefore identification and authorization control of the NIST 800-53 describes relevant controls to this layer of an SDN infrastructure.

- IA-3 Device identification and authentication: a bidirectional cryptographic authentication method such as TLS must be implemented for all traffic between the SDN controller and the Data plane. This would avoid implementation of any rogue device in the Data plane.
- IA-4 Identifier management: an identification manager can be used to identify each device on the Data plane.
- IA-7 Cryptographic module authentication: all communication between data plane and SDN controller needs to be mutually authenticated using FIPS approved message authentication code algorithm

D. Data plane

- IA-7 Cryptographic module authentication: authentication of each device at the Data plane is required prior to any communication.
- SC-5 Denial of service protection: all switches must be configured to minimize the amount of traffic directed to the SDN controller.

The proposed controls from the four groups above are implemented in the existing SDN implementations to certain extent. To test the proposed set of controls, four of the well-known open source SDN controllers - OpenDayLight, ONOS, Floodlight, and Ryu SDN controller were used to show

applicability of the proposed methodology in the running SDN environment.

IV. Experiments and Findings

The objectives of the experiment performed in the this section was to simulate a running SDN infrastructure and test the evaluation methodology proposed in the previous section. Since the methodology sets the minimum-security baseline, checking the existence of the required controls and their default settings would allow security analyst to assign security score to the specific SDN controller which is being tested.

To evaluate selected SDN controllers, the following network topology was used. The simulation of the network was done with VMware software used as the VM manager and three different virtual machines were created as shown in the figure 2.

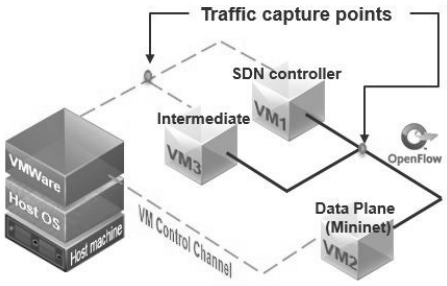

Fig. 2. Lab Topology

The SDN controller is installed on VM1 and the simulated Data plane is done on VM2 where a network topology simulator (Mininet) is installed. A third VM was used to perform attacks and capture traffics between a) the SDN controller and the Host machine which represents the Application plane; b) between the SDN controller and VM1 the Data plane. The three VMs have a bridged NIC which enables them to be on the same subnetwork as the host machine and communication between the three VMs and the host is done through the VM control channel (VMware switch). Traffic between the VM1 and VM2 uses OpenFlow v. 1.0 protocol. VM3 is also able to tap in between the OpenFlow channel created at the capture points. Inside Mininet, a basic network was created with four OpenFlow switches, with 2 hosts per switch. The subsections below outline experiments on security evaluation of simulated SDN infrastructures.

A. Application Plane and Northbound Communication

At the Northbound communication path various tests have been performed to deter-mine security mechanisms in place. A comparison of the security mechanisms in place with the proposed metrics was performed afterwards.

Description of Experiments: a) To check existence and FIPS compliant authentication process at the Application Plane; b) To check if traffic at the northbound layer is encrypted - this test was done by capturing traffic from the intermediate machine. c) If authentication is bypassed, is there an alternate security mechanism to avoid illicit flow rule insertions from rogue applications; d) To verify if there is a process responsible for keeping accurate logs of every transaction at the northbound layer.

```
  Host: 192.168.232.131:8181\r\n
  Connection: keep-alive\r\n
0. Content-Length: 0\r\n
  Accept: application/json, text/plain, */*\r\n
  Origin: http://192.168.232.131:8181\r\n
  User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
• Authorization: Basic YWRtaW46YWRtaW4=\r\n
    Credentials: admin:admin
  Content-Type: application/yang.data+json\r\n
Referer: http://192.168.232.131:8181/index.html\r\n
```

Fig. 3. Credentials are being transmitted in plain on OpenDayLight.

Findings: The results from the security evaluation performed on selected SDN environments is discussed in this section. It was noted that there is an authentication method put in place before any communication at the Northbound layer, but due to the lack of encryption, username and password used to authenticate application can be collected by a malicious device inserted on the network. Ryu and Floodlight controllers do not provide any authentication for applications by default (Fig. 3).

Results collected from packets capturing on the Intermediate VM has revealed there is no encryption in place to ensure confidentiality of data transferred between the SDN controller and the Application layer on all SDN controller tested. On all SDN tested the entire network topology can be inferred from captured traffic.

With the credentials collected a topology table request was send from the inter-mediate VM. On OpenDayLight and Floodlight SDN the SDN controller authenticated the malicious request as authentic from a valid application but on both ONOS and Ryu, a reply was send back to the Intermediate VM as unauthorized operation.

In order to avoid eavesdropping and rogue applications, all application must be authenticated by the controller prior to any exchange of data. A comparison of all the available features for OpenDayLight and the initial settings has shown that the activity logging was not enabled by default. It is recommended to enable monitoring on any SDN to establish accountability.

B. Control Plane

Description of Experiment: Check points at this layer depend on the environmental setup of the organization. DoS attack was simulated to check if there is an appropriate DoS mitigation mechanism on the SDN controllers tested. Each SDN controller tested was monitored after a series of large traffic bursts sent from the Data plane.

Findings: A DoS attack was launched from the Data plane on the controller to see how the controller reacts when there is a large amount of traffic coming from the data plane. An increase of network load was noticed from all the controllers and this has also created latency in the network. In order to mitigate DoS on the controller, flow control application needs to be deployed to control the amount of traffic which goes to the controller. After implementing sFlow on OpenDayLight and ONOS, the DoS attack was launched again and sFlow was

able to detect and block packets related to the flooding attack. Floodlight and Ryu controllers can contain the DoS attack.

C. Southbound communication layer

Description of Experiment

At this layer traffic has been captured to determine if an appropriate encryption is used. List of enabled features by default was also checked to verify existence of a device identification module.

Findings

For all SDN controllers tested, there is no Encryption done at the southbound communication layer. List of enabled features by default on all the Tested SDNs has demonstrated that there is no device identification and authentication done at this layer. On OpenDayLight and ONOS controller the netconf:identifier feature needs to be activated by default. All communications must be done using TLS to ensure confidentiality.

D. Data Plane

Description of the experiment:

Configuration of the Data plane depends on each organization. At this layer one test has been conducted to determine if devices on the Data plane are authenticated by the controller prior to any communication. Since the device at this layer are virtualized using a network simulator, tests conducted at this layer did not cover all security requirement for this layer.

Findings:

Each SDN controller tested identifies and authenticates all devices connected from the Data plane by their mac address, and Switch mac address prior to any communication. The effectiveness of the authentication module in place was not tested in this research since all these devices are been virtualized. In a physical implementation it is recommended to also test effectiveness of the authentication module in place at the Data plane. Mininet network simulator provides basic OpenFlow switch functionalities which does not have DoS protection. This module was also not tested in this research.

TABLE II. SUMMARY OF SDNS SECURITY ASSESSMENT RESULT. X MEANS TEST FAILED, ✓ MEANS TEST PASSED

Module tested	ODL	ONOS	Floodlight	Ryu
IA-3	✓	✓	X	X
IA-7 Northbound communication	X	X	X	X
SC-8	X	X	X	X
AU-2 AU-10	X	X	✓	✓
SC-5	X	X	✓	✓
IA-7 Southbound communication	X	X	X	X
IA-7 Data plane	✓	✓	✓	✓

Out of the 16 defined evaluation checkpoints, 7 were tested on the simulated environments. Since the evaluation environment was simulated for this test, some controls cannot be evaluated because the environment does not meet an organization defined environment. An example of one control that was not tested is SA-2 which talks about allocation of enough computing resources to the physical machine hosting the SDN controller which can vary based on the needs of the organization. From the results obtain, it can be noticed that with default configuration, OpenDayLight and ONOS meets only two security controls, Ryu and Floodlight meets three. Based on the organization's defined security policy and the importance of each metrics, a score can be assigned to each evaluation points to determine a general score for each SDN infrastructure tested. The general score determined can then be used to classify the SDNs in term of security.

It is important to stress that the experiment did not have an aim to compare the security posture of the 4 SDN controllers. The aim was to show that the proposed methodology comprehensively covers required controls listed in NIST 800-53 document.

V. CONCLUSIONS AND FUTURE WORKS

The Security Evaluation Methodology described in this paper enables network security expert to classify available SDNs in terms of security. The discussion and results outlined in the paper can help organizations to perform structured SDN security evaluation and based on the results, decide which SDN infrastructure to adopt. Pro-posed metrics defined in this paper can serve as a guideline to patch security vulnerabilities identified in the designated SDN infrastructure. This research provides comprehensive way for all organization to assess security in any SDN infrastructure in contrast with SDN security assessment tools that can only be used to test specific set of SDN controllers under a predefined environment that can be different from the organization defined environment. Results of the experiments have attested useful-ness and applicability of the proposed methodology. One might consider improving on the proposed methodology by automating all the controls evaluation with, for example, SCAP rules.

REFERENCES

[1] S. J. Vaughan-Nichols, "Virtualization Sparks Security Concerns," Computer vol. 41, 2008.

[2] T. Dargahi, A. Caponi, M. Ambrosin, G. Bianchi and M. Conti, "A Survey on the Security of Stateful SDN Data Planes," IEEE Communications Surveys & Tutorials, vol. 19, no. 3.

[3] A. Feghali, R. Kilany and M. Chamoun, "SDN security problems and solutions analysis," in Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS), Paris, 2015.

[4] U. C. Framework, "Network Security Requirements Guide," Unified Compliance Framework, [Online]. Available: https://www.stigviewer.com/stig/network_security_requirements_guide/.

[5] O. N. Foundation, "software-defined-standards specifications," Open Networking Foundation, 2018. [Online]. Available: https://www.opennetworking.org/software-defined-standards/specifications/.

[6] R. S. Ross, "Security and Privacy Controls for Federal Information Systems and Organizations," Special Publication (NIST SP)-800-53 Rev 4, 2013.

[7] J. Medved, R. Varga, A. Tkacik and K. Gray, "OpenDaylight: Towards a Model-Driven SDN Controller architecture," 2014 IEEE 15th

International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)(WOWMOM).

[8] B. Pankaj, G. Matteo, H. Jonathan, H. Yuta, K. Masayoshi, K. Toshio, L. Bob, O. Brian, R. Pavlin, S. William and P. Guru, "ONOS: towards an open, distributed SDN OS," In Proceedings of the third workshop on Hot topics in software defined networking (HotSDN'14).

[9] B. S. Networks, "Floodlight," [Online]. Available: http://www.projectfloodlight.org/floodlight/.

[10] SDX central "What is Ryu Controller,", [Online]. Available: https://www.sdxcentral.com/sdn/definitions/sdn-controllers/open-source-sdn-controllers/what-is-ryu-controller/.

[11] A. Thongthua and S. Ngamsuriyaroj, "Assessment of Hypervisor Vulnerabilities," Cloud Computing Research and Innovations (ICCCRI), pp. 71-77, 4-5 May 2016.

[12] J. Sahoo, S. Mohapatra and R. Lath, "Virtualization: A survey on concepts, taxonomy and associated security issues," in Computer and Network Technology (ICCNT), 2010.

[13] R. Anand, S. Sarswathi and R. Regan, "Security issues in virtualization environment," in Radar, Communication and Computing (ICRCC), Tiruvannamalai, 2012.

[14] S. Natarajan and T. Wolf, "Security issues in network virtualization for the future Internet," in Computing, Networking and Communications (ICNC), 2012.

[15] A. Kumara and C. D. Jaidhar, "Hypervisor and virtual machine dependent Intrusion Detection and Prevention System for virtualized cloud environment. In Telematics and Future Generation Networks (TAFGEN)," IEEE, pp. 28-33, 2015.

[16] M. Pearce, S. Zeadally and R. Hunt, "Virtualization: Issues, security threats, and solutions," in Computing Surveys (CSUR), New York, 2013.

[17] P. Sheinidashtegol and M. Galloway, "Performance Impact of DDoS Attacks on Three Virtual Machine Hypervisors," in Cloud Engineering (IC2E), 2017 IEEE International Conference on, Vancouver, 2017.

[18] I. Ahmad, S. Namal, M. Ylianttila and A. Gurtov, "Security in Software Defined Networks: A Survey," IEEE Communications Surveys & Tutorials, pp. 2317 - 2346, 27 August 2015.

[19] B. Kevin, L. J. Camp and C. Small, "Openflow vulnerability assessment," Proceedings of the 2nd ACM SIGCOMM workshop on Hot topics in software defined networking,

[20] J. Martins, M. Ahmed, C. Raiciu and F. Huici, "Enabling Fast, Dynamic Network Processing with ClickOS," roceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, pp. 67-72, August 2013.

[21] B. Carpenter and B. Scott, "Middleboxes: Taxonomy and issues," RFC 234, 2002.

[22] S. Mudit and K. Rakesh, "A recent trends in software defined networking (SDN) security," in Computing for Sustainable Global Development (INDIACom), New Delhi, 2016.

[23] G. Uttam, C. Pushpita, T. Deepak, S. Sachin, X. Kaiqi and K. Charles, "An SDN Based Framework for Guaranteeing Security and Performance in Information-Centric Cloud Networks." Int. Conference on Cloud Computing (CLOUD), Honolulu, 2017.

[24] U. C. Framework, "Network Security Requirements Guide," Unified Compliance Framework, [Online]. Available: https://www.stigviewer.com/stig/network_security_requirements_guide/.

[25] S. Scott-Hayward, C. Kane and S. Sezer, "OperationCheckpoint:SDN Application Control," in 22nd International Conference on Network Protocols, Raleigh, 2014.

[26] P. Phillip, S. Seungwon, Y. Vinod, F. Martin and S. Keith, "Securing the Software-Defined Network Control Layer," in Network and Distributed System Security Symposium, 2015.

[27] A. Dixit, F. Hao, S. Mukherjee, T. Lakshman, R. Kompella, P. University and B. L. Alcatel-Lucent, "Towards an elastic distributed SDN controller," Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, pp. 7-12, August 2013.

[28] H. Yeganeh, Soheil and Y. Ganjali, "Kandoo: a framework for efficient and scalable offloading of control applications.," Proceedings of the first workshop on Hot topics in software defined networks, pp. 19-24, 2012.