Security evaluation methodology for Software Defined Network solutions

Co-authored by

Student: Djahlin Jean-Claude Nikoue

Primary advisor: Sergey Butakov

Secondary advisor: Yasir Malik

Project report

Submitted to the Faculty of Graduate Studies,
Concordia University of Edmonton

In Partial Fulfillment of the
Requirements for the Final
Research Project for the Degree

MASTER OF INFORMATION SYSTEMS SECURITY

MANAGEMENT

Concordia University of Edmonton

FACULTY OF GRADUATE STUDIES

Edmonton, Alberta

April 2018

Security evaluation methodology for Software Defined Network solutions

Djahlin Jean-Claude Nikoue, Sergey Butakov, Yasir Malik
Department of Information Systems Security and Assurance Management
Concordia University of Edmonton, T5B4E4,
Edmonton, Alberta, Canada
jnikoue@csa.concordia.ab.ca {sergey.butakov, yasir.malik }@concordia.ab.ca

***Abstract*— Software Defined Networking (SDN) a novel approach to networking has introduced both innovative opportunities and disadvantages in the networking field. The opportunities brought by this technology varies from the facility in configuring and managing a vast and dynamic network while using less resources and time to the ability to apply an intelligent and dexterous network security mechanism against malicious flows without the use of a specialized network security hardware. Even though this novel technology seems to promise a lot of advantages, it nonetheless comes with various vulnerabilities which can be associated with both virtualization and the traditional approach to networking. There is a variety of SDN controller providers on the market for organizations but each of them comes with security flaws that are either unique or common to SDNs, which makes deciding on which SDN to implement a tough decision for network professionals. This research proposes to deliver a comprehensive way for organization to evaluate security vulnerabilities in SDN infrastructures which will serve as a guideline while deciding which SDN to adopt. The vulnerability assessment proposed in this research is layered to evaluate each layer of the SDN architecture and each evaluation metrics defined in this research has been matched from the security controls defined in NIST 800-53. The security evaluation methodology proposed has also been tested and result from the test is also documented to provide more comprehensiveness.**

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

I. INTRODUCTION

In the quest to adopt a Software Defined Networking (SDN), one of the major concern for an organization would ask is how secure is it. SDN which is a novel approach to networking not only comes with major vulnerabilities identified in a traditional networks and virtual systems but also with specific vulnerabilities especially the ones associated with the SDN Control plane. Although Security Technical Implementation Guide (STIG) [1] and Open Network Foundation (ONF) [2] have placed some security controls and principles to achieve a secure SDN infrastructure, decision is left to the organization that wants to deploy an SDN to verify compliance to these controls before choosing which SDN infrastructure suits best their needs. With the increasing number of SDN solutions, providing a security evaluation methodology for SDNs will help in deciding which SDN to adopt in an organization from a security viewpoint.

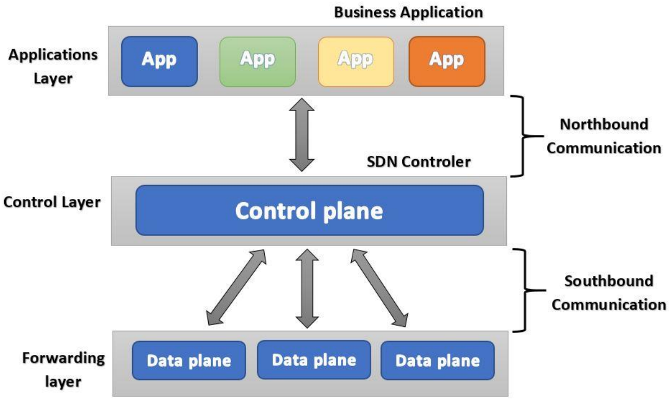


Figure 1 SDN architecture

In this paper, an SDN security evaluation methodology that will serve as a guideline for evaluating security implementations in an SDN infrastructure. The methodology covers all aspects of the SDN architecture from Application plane to Data plane. Each evaluation metric has been defined from list of common vulnerabilities specific to SDNs infrastructure and system virtualization and each metric is mapped from security controls defined in NIST 800-53 and the STIG for SDNs. To justify the proposed methodology, various SDN controllers such as ONOS, HPE and OpenDayLight have been implemented and tested in a simulation environment with a network simulator (Mininet) used to create the tested network. Furthermore, analysis of the results obtained from the tested environment has been interpreted and for some vulnerabilities found, mitigations defined in STIG for SDN has been proposed to address those vulnerabilities.

II. REVIEW OF RELATED WORK

Virtualization has improved resource utilization and expanded scalability by simulation of hardware resources and decoupling software applications or services confined to run on a computer system [4]. Improvement work done on virtualization technologies has led to Software Defined Networking(SDN) - a networking paradigm in which the control and management of the network is separated from the traffic forwarding primitives [5]. The management of the network is done at the Application Plane, the traffic control is centralized in a single device called the Control Plane, separated from the traffic forwarding devices which are in the Data plane. SDN technology is being implemented by the top technology companies at an increasing rate. Google who started the implementation of SDN in 2010 has already 25% of its traffic using an SDN infrastructure in 2016 [6]. Google could reach almost 100% network utilization with their SDN in opposition to the traditional architecture that allows only 30% to 40% utilization [7]. Separating the Data plane and the Control plane provides an SDN network enough resilience to merge the advantages of system virtualization and cloud computing [8]. However, this new approach added to networking does not come with just benefits but brings extra challenges as well.

Concerns raised by virtualization vary from performance to security. Those security concerns can be found as follows:

- Problems with Hypervisor: VM escape [9] [10]; Single point of failure [10] [11], improper input validation [12], failure to maintain processes within the bounds of memory buffer [12], [11]; improper authentication [12], [11], [13], and authentication bypass by capture-replay [12], [14]

  The above researches demonstrate how critical the hypervisor is and how its failure constitutes a single point of failure to the entire virtualized infrastructure.
- Problem with communication: DoS attack on the host machine [10], [15]; Unauthorized access to network due to inappropriate authorization [12], [14], [13].

In [12] security issues related to Virtualized networks have been compiled with scenarios into issues related to virtualized networks, the network infrastructure, and the users of the virtual network. The limitations of both research [10] and [12]is that the researchers proposed few solutions to the most know vulnerabilities in a virtualization environment but do not provide a detailed way to evaluate security in a virtual machine internetworking.

Focusing on only the issues related to SDNs, research has shown that from a fundamental point of view, 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 [16].The vulnerabilities list identified in [16] in addition to proposed solution from other researchers is compiled in table 1. At the Data plane, security vulnerabilities can be associated to the protocols governing 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. Researchers in [8] have identified absence of encryption of the traffic between the Control plane and forwarding layer and a weak authentication which can lead to attacks such as Man-in-the-Middle (MiM). Researches in [17] have proposed possible solution to the vulnerabilities identified and highlighted the limitations to those solution described in table 1. Researchers in [18] proposed ClickOS which is a light operating system that was built with only the libraries required for the application that runs on it (unikernel). This unikernel system based runs on Xen hypervisor. The proposed solution helps create middleboxes, a system created to increase security trough packet filtering, Intrusion detection and prevention, and increase performance with functionalities such as Proxying, protocol acceleration and WAN optimization [19]. The proposed middlebox reinforces security and improve isolation of each devices on the forwarding plane. KANDOO framework a proposed solution in [20] suggests an implementation of a distributed Control plane instead of having a single Control plane. At the other hand FRESCO framework, another proposed solution focuses more on the packet filtering rules and securities policies rather than the low-level security of OpenFlow. Researchers in [21] proposed to have security of the SDN distributed between various applications such as an Abstraction layer where the Control plane resides, and which control the other managers, a Cache Manager which is implemented at the Data plane to keep record of all traffics, a Routing Manager responsible to identify available routes and keep the Control manager updated with routes, a Security Manager responsible for data encryption and integrity of data across the SDN, Policy/QoS Manager where the firewall rules are sets, a Virtualization Manage which contains the architecture of the virtual machines connected to the network and a Naming Manager responsible for naming each device in the network infrastructure. The proposed framework is more focused on performance and availability rather than on other aspects of security such as confidentiality and integrity. A further research can be done to develop a framework that covers all aspect of security in an SDN solution.

Table 1 outlines a list of vulnerabilities identified by previous researches. This list of vulnerabilities is structured following an SDN architecture. The table starts from the Application Plane down to the Control Plane and then down to the Data Plane and communication between each of the planes is also covered. The northbound communication represents traffic between the Application Plane and the Control Pane, and the Southbound communication represent communication between the Control Plane and the Data Plane. The list of vulnerabilities identified is mapped in the table to some known security controls to each of them and the limitations to the implementation or effectiveness of those security controls.

Review of previous work has demonstrated that researchers after identifying vulnerabilities in virtualized environment and SDN environments provided a number of useful recommendations, but most of the recommendations are directed to SDN developers in the quest to build secure SDN and network architects who are deploying an existing SDN. Research done in [8] and [18] as shown in table 1 proposes various solutions to known vulnerabilities to SDN but those researches do not provide a way to verify if recommended solutions have been implemented. Limited research has been done to assist IT management and network professionals in evaluating security in various SDN solutions. Such advices are required to provide management with advice on the suitability of the SDN solutions for organization’s needs from the security perspective. The research work bellow focuses on developing security assessment methodology that will help network professionals in assessing security in various SDN solutions in order to decide which SDN solutions suits best their needs from the security standpoint [1].

Table 1 list of vulnerabilities, proposed solutions, and limitations

Layer	Vulnerabilities identified	Controls	Potential control limitations
Application Plane	Lack of authentication and authorization of applications [16]	Use of a Security Enforced Kernel (SEK) like SE Floodlight to authenticate applications and detects rules conflicts [22]	Limited to Floodlight controller and requires and administrator to pre-sign the applications java class
	Lack of access control and accountability [16]		
	Lack of application isolation can lead to inconsistent flow rules [10]	Check flow rules contradiction in real time and implement role-based authorization through a security enforcement kernel [23]	Requires complex algorithm to determine application security level.
Northbound Communication	Fraudulent flow rules insertion due to non-verification of the application by the controller [16]		
	Weak authentication between the applications and the controller, which may lead to spoofing attack by spoofing northbound messages [8]	Use of TLS for encryption: avoid eavesdropping and spoofed communications, and validate the identity of each component [8]	It is optional and very limited number of vendors supports it. And as proposed, no complete easy to use infrastructure is considered [8]
Control Plane	DoS Attacks [16] [8]	Rerouting traffic to a middlebox [8], [18]	Temporary hybrid solution, since it limits the flexibility and scalability of SDN [8]
	Implementation of rogue controller to edit flow entries to put the entire SDN under attacker's control [8]	Hardening and monitoring of the controller [8]	Depends on the implementation and on the underlying operating system [8]
	Threats from unauthorized Controller access [16]		
	Controller hijacking or compromise [16]		
	Threats due to Scalability [16]	Use of Distributed controller [24], [25]	Can increase latency in the network
Southbound Communication	Eavesdropping and spoofing possible due to no encryption of communication between controller and switches	Use of Security manager to encrypt communication controller and switches [21]	May not be easy to implement since requires implementation of a PKI at the controller level and can increase latency
	Weak authentication can lead to Man-in-the-Middle attack at this level [8]	Use of a naming manager at the controller to name each device on the network and authenticate them before communication is established [21]	Limits flexibility and scalability because naming is done at the initial stage of the network before any communication is established.
	Unauthorize access to network due to inappropriate authorization [8]		
Data Plane	Flooding attacks [16]	Configure switches to be able to filter out illicit connections using SYN cookies and a special TCP handshake procedure [20]	Requires additional configuration of each switches on the entire SDN network which defeats the purpose of a centralized controller (SDN).
	TCP-Level attacks [16]		

III. Descriptyion of the Experiment

The objective of the experiments performed is to simulate a running SDN infrastructure and perform a security evaluation on the simulated environment to check if the selected SDN controllers follow the minimum standard required in the quest to achieve a secure network infrastructure. Various data capture and attacks will be performed at different points in the network. The compiled result from the simulations in addition to the evaluation metrics will testify usefulness of the proposed research.

To evaluate selected SDN infrastructures, a network topology was put in place. The network topology was simulated in a virtual environment with VMware software used as the VM manager and three different virtual machines where created as depicted in the figure 2. bellow

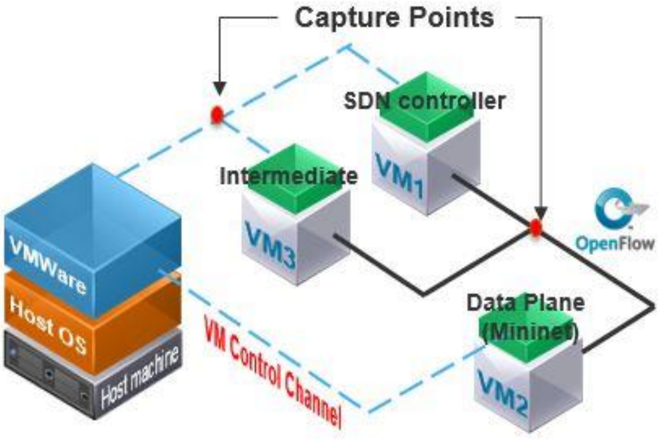


Figure 2 lab Topology

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 is used to perform attacks and capture traffics between the SDN controller and the Host machine and between the SDN controller and 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 version 1.0 of OpenFlow protocol. VM3 is also able to tap in between the OpenFlow channel created at the capture points. Inside Mininet, the network simulator used in this research, a basic network was created with four OpenFlow switches, height hosts, 2 host per switch. The simulated network was then linked to a remote SDN controller by its IP address.

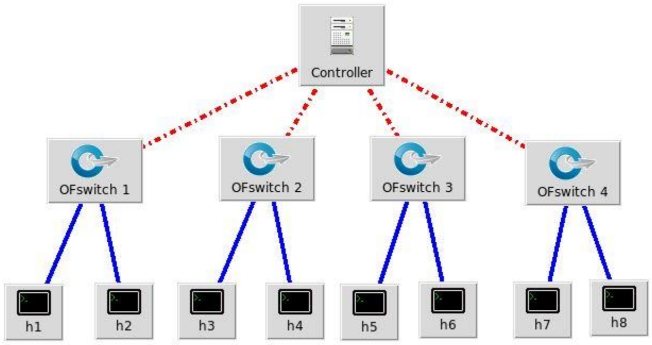


Figure 3Mininet network

Security Evaluation of Simulated SDN infrastructures

A. Application Plane and Northbound Communication

At the Northbound various test has been performed to determine security mechanism in place on the simulated environments.

IV. SECURITY EVALUATION METHODOLOGY

Based on the review of related works, enough information can be deduced to define metric that need to be evaluated on an SDN infrastructure.

This metrics can be classified from the Top layer of an SDN, Application plane to the lowest layer, the Data plane.

A. Application Plane and Northbound Communication

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

- Each application needs to be authenticated prior to sending any instruction.
- In addition to Authentication, an application manager needs to be implemented such as a security enforced kernel to avoid having more than one application doing the same role.
- All communication between Northbound APIs and SDN controller needs to be mutually authenticated using Federal Information Processing Standard (FIPS)-approved message authentication code algorithm.
- Communication between the Application Plane and the Control plane must be encrypted to ensure confidentiality of data share between those two layers.
- 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 traffics from the Data plane.

- The SDN controller must be deploy on a dedicated computer with enough computing resources to handle all traffic.
- To avoid the issue of single point of failure, a cluster of SDN controllers will be required.
- A flow control application need to be deployed to reduce the control the amount of traffic sends to the controller and detect possible DoS and DDoS attacks.
- The Physical computer hosting the SDN must be dual homed with at least two Network Interface Cards (NICs) with link aggregation implemented on the interfaces connecting to the Data plane in order to ensure high availability.
- A Host Intrusion Detection System (HIDS) needs to be implemented on the machine hosting the SDN controller.

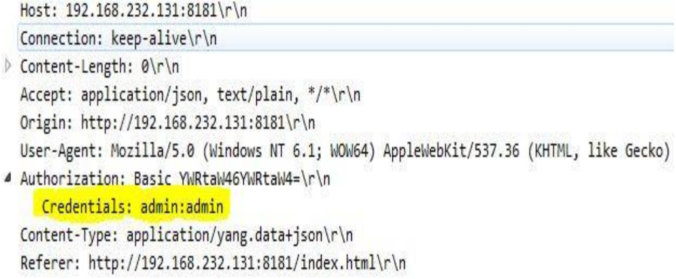
V. FINDDINGS AND DISCUSSIONS

The results from the security evaluation performed on selected SDN environments is discussed in this section.

A. Application Plane and Northbound Communication.

Results collected from packets capturing from Wireshark on the Intermediate VM has revealed for all three tested SDNs, there is no encryption algorithm in place to ensure confidentiality of data transferred between the SDN controller and the Application layer. On both ONOS and OpenDayLight SDN the entire network topology can be inferred when there is a network topology request send from the topology manager application.

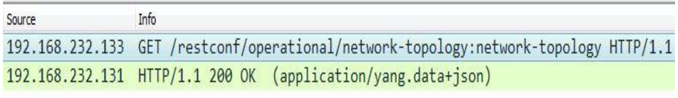
It was noted that there is an authentication method put in place before any communication at the Northbound layer, but due to the lack of encryption, username and password used to authenticate application can be collected by a malicious device inserted on the network as shown in the figure bellow.



```
Host: 192.168.232.131:8181\r\n
Connection: keep-alive\r\n
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
```

Figure 4 collection of apps credential using Wireshark

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



Source	Info
192.168.232.133	GET /restconf/operational/network-topology:network-topology HTTP/1.1
192.168.232.131	HTTP/1.1 200 OK (application/yang.data+json)

Figure 5 malicious topology requests authorized

VI. Conclusions and Future works

The discussion and results outlined in the paper can help organizations to perform a structured SDN security evaluation, and based on the results after analysis decide which SDN infrastructure to adopt. The Security evaluation methodology described in this paper enables network security expert to classify SDNs in terms of security. Proposed metrics defined in this paper serves as a guideline to patch security vulnerabilities identify on a designated SDN infrastructure. The security evaluation methodology can be tested again to verify compliance to required security recommendation. One can add more test scenarios to the security evaluation methodology in this paper and with more research being done to improve on SDNs additions security recommendations can be further added to the methodology.

VII. References

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

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

[3] S. Verlrajan, "SDN Architecture," The Tech, 25 December 2012. [Online]. Available: http://www.thetech.in/2012/12/sdn-architecture.html. [Accessed september 2017].

[4] S. J. Vaughan-Nichols, "Virtualization Sparks Security Concerns," in *Computer volume 41*, 2008.

[5] 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, pp. 1701 - 1725, 30 March 2017.

[6] J. Rexford, A. Vahdat and D. Clark., "A Purpose-Built Global Network: Google's Move to SDN," *Communications of the ACM,,* vol. 59 No. 3, pp. 46-54, 2016.

[7] J. Dix, "A Q&A with a principle engineer on the motivations for going with OpenFlow, the learnings, what's next," 07 june 2012. [Online]. Available: https://www.networkworld.com/article/2189197/lan-wan/google-s-software-defined-openflow-backbone-drives-wan-links-to-100--utilization.html. [Accessed 14 september 2017].

[8] 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.

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

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

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

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

[13] 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.

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

[15] 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.

[16] 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.

[17] B. Kevin, L. J. Camp and C. Small, "Openflow vulnerability assessment," *In Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking,* pp. 151-152, August 2013.

[18] 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.

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

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

[21] 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." In Cloud Computing (CLOUD)," in *International Conference on Cloud Computing (CLOUD),* Honolulu, 2017.

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

[23] 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*, San Diego, 2015.

[24] 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.

[25] 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.

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

[27] P. Phillip, S. Seungwon, Y. Vinod, F. Martin, T. Mabry and G. Guofei, "A Security Enforcement Kernel for OpenFlow Networks," *SRI International*, pp. 121-126, 13 AUgust 2012.

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