# Enhancing Security Education Through Designing SDN Security Labs in CloudLab

Younghee Park
San Jose State University
younghee.park@sjsu.edu

Hongxin Hu
Clemson University
hongxih@clemson.edu

Xiaohong Yuan
North Carolina A&T State University
xhyuan@ncat.edu

Hongda Li
Clemson University
hongdal@clemson.edu

## ABSTRACT

Software-Defined Networking (SDN) represents a major shift from ossified hardware-based networks to programmable software-based networks. It introduces significant granularity, visibility, and flexibility into networking, but at the same time brings new security challenges. Although the research community is making progress in addressing both the opportunities in SDN and the accompanying security challenges, very few educational materials have been designed to incorporate the latest research results and engage students in learning about SDN security. In this paper, we presents our newly designed SDN security education materials, which can be used to meet the ever-increasing demand for high quality cybersecurity professionals with expertise in SDN security. The designed security education materials incorporate the latest research results in SDN security and are integrated into CloudLab, an open cloud platform, for effective hands-on learning. Through a user study, we demonstrate that students have a better understanding of SDN security after participating in these well-designed CloudLab-based security labs, and they also acquired strong research interests in SDN security.

## KEYWORDS

Software-Defined Networking, Security, CloudLab

**ACM Reference Format:**
Younghee Park, Hongxin Hu, Xiaohong Yuan, and Hongda Li. 2018. Enhancing Security Education Through Designing SDN Security Labs in CloudLab. In *Proceedings of The 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3159450.3159514

## 1 INTRODUCTION

Software-Defined Networking (SDN) has progressed from a purely theoretical concept [10, 11, 13, 18] to an increasingly popular new paradigm that numerous networking vendors are not only embracing, but also are pursuing as their model for future enterprise network management. According to a recent report from Google, SDN-based network management helped Google run wide-area networks at close to 100% utilization compared to other state-of-the-art network environments that yielded only 30% to 40% network utilization [16].

Along with the rapid growth of SDN, there are many new security issues in SDN that are different from traditional network security problems [7, 9, 12, 14, 15, 17, 19, 26, 27]. It is imperative that research results on SDN should be brought into cybersecurity education with effective delivery approaches to prepare our students, as the future workforce and pillars of the nation, with security awareness and readiness sufficient to protect the emerging SDN-based network infrastructure. However, there are few hands-on lab materials to address SDN security issues. A traditional lab setting with multiple interconnected physical computers for cybersecurity training presents problems both in terms of associated overhead costs and in terms of setup, configuration, installation, and scheduling and management of equipment [22]. Usually, the SDN learning environment is also limited to simulation-based exercises. Furthermore, commercial cloud platforms like Amazon Web Service (AWS) are unflexible and expensive to use, and have restrictions on conducting some security labs [22]. In addition, no existing *commercial* cloud platforms can directly provide SDN features that are critical for implementing SDN security labs.

This paper presents our initial progress to develop Cloud-based SDN security labs. We develop an open laboratory for SDN security education, leveraging unique features provided by CloudLab [1], an open cloud platform. CloudLab is specially built for research and education purposes and hence it is completely *free* to use. In addition, CloudLab supports OpenFlow [18] and other software-defined networking technologies that are particularly important for building SDN security labs. It engages students in SDN security education through hands-on lab development. Our designed labs cover various security issues, such as denial-of-service (DoS) attacks and application attacks, from the data plane to the control plane in SDN. We also introduce pilot labs related to two main security problems in SDN and students participate in survey after finishing the labs. Our survey results show that the designed labs provide students with a positive learning experience in SDN security.

The rest of this paper is organized as follows. Section 2 introduces CloudLab. Section 3 presents various SDN security labs, and also gives example lab modules. Section 4 discusses student experiences in the proposed SDN security labs in CloudLab. Section 5 reviews related work and lastly Section 6 concludes this paper.


*SIGCSE '18, Feb. 21–24, 2018, Baltimore, MD, USA*

ACM ISBN 978-1-4503-5103-4/18/02…$15.00
https://doi.org/10.1145/3159450.3159514

## 2 CLOUDLAB FOR SDN SECURITY EDUCATION

SDN as an new emerging network technology has raised a lot of new research problems, especially in security areas. Even though SDN has rapidly adopted in both industry and academia with significant research outcomes, there are very few education materials from the latest research results in order to effectively teach students in SDN security. Therefore, we aim to build a cloud-based open laboratory for hands-on labs in SDN security.

Several studies [22, 29, 31] have demonstrated the use of cloud computing as an effective delivery approach for cybersecurity education; unfortunately their emphasis involves leveraging *commercial* cloud platforms, such as Amazon Web Services (AWS), which are expensive to use and have restrictions on conducting some security labs [22]. More importantly, no existing commercial cloud platforms can directly provide SDN features that are critical for conducting our hands-on labs on SDN security.

CloudLab is an NSF-sponsored project that enables the academic research community to develop and experiment with novel cloud architectures and to develop new cloud computing applications. CloudLab provides the substrate on which instructors can build their own flexible cloud environment that will give them both control and visibility all the way down to the hardware.

CloudLab has some specific advantages for building an open laboratory for SDN security education. First, unlike the creation of security exercises using the commercial cloud, which is quite costly, CloudLab was designed for educational purposes, which means that it is *free* of charge. Second, CloudLab supports OpenFlow (an open SDN standard through which instructors may run experimental protocols on campus networks) and other SDN technologies. This is significant requirements for SDN security education. In the case of exclusive-access bare metal switches, users will get direct and complete OpenFlow access to switches to many SDN security experiments. Third, CloudLab uses a profile to encapsulate everything needed to run an experiment. The profile consists of two main parts: a description of hardware, storage, and network resources needed to run the experiment, and software artifacts that run on those resources. A profile can be created and published by an instructor. Students use the profile to provision an entire cloud inside of CloudLab within minutes, which greatly reduces the set-up time for experiments, making it easier to repeat them and compare results. Last, existing research [22] has confirmed that unlike CloudLab, the Amazon AWS cloud or other commercial cloud alternatives may be unsuitable for many cybersecurity labs, such network security exercises involving eavesdropping and Man-In-The-Middle (MITM) attacks.

We leverage the unique benefits provided by CloudLab to build an *open* laboratory specifically for the SDN security education purpose. Using the CloudLab-based open laboratory, an instructor designs and publishes an experiment simply by creating a profile and distributing it to the students. There are two ways to create a profile: 1) cloning an existing profile, which is useful if the instructor wants to design an experiment using a previous record, and 2) writing a new profile in RSpec format [2]. CloudLab provides two methods [3] to easily generate valid profiles in RSpec format. The graphical user interface (GUI) method is straightforward, based on Jacks, an embeddable RSpec editor. The programming language binding method is useful when the user wants to create large, complicated profiles in the RSpec format by writing programs in an existing, popular language (like Python). Geni-lib [4] is being evaluated for this purpose.

## 3 DESIGN OF SDN SECURITY LABS

### 3.1 Lab Module Overview

The SDN security labs designed with the support of CloudLab consist of five lab modules to learn five important security problems in SDN. Each module has a problem definition to address a specific security issue in SDN, a learning objective and outcome to evaluate student achievement, and research challenging questions to encourage students to participate in research in the specific SDN security issues.

#### 3.1.1 **Lab 1***: Data-to-control Plane Saturation Attacks to the SDN Controller*

. Problem Definition: Data-to-control plane saturation attacks present the biggest security threat to the SDN controller because the controller is a logically centralized framework [24, 33, 34]. The lack of scalability can cause control plane saturation attacks by inundating communication between the controller and the switch. When a table-miss flow entry occurs in an OpenFlow switch, the packet is forwarded to the OpenFlow controller. The controller responds with one or more flow rules for packet processing actions. Such design in SDN becomes a scaling bottleneck since anomalous burst traffic, such as flash crowds, denial-of-service attacks, and even botnets, quickly saturate the control plane with new flow requests. An adversary could exploit the feature by mounting a control plane saturation attack that disrupts network operations in SDN.

Learning Objectives: Students will learn the communication logic between the controller and the data plane as to connection establishment and flow rule installation.

Learning Outcomes: Students will be able to understand fundamental security issues in the controller. They can generate TCP-based flood attacks on the controller, and analyze flow rules and response delays under attack.

Challenging Questions: Students will brainstorm how to increase scalability and resilience in the controller. Students will discuss pros and cons of using multiple controllers to prevent or defeat saturation attacks [33].

#### 3.1.2 **Lab 2***: Flooding Attacks to the SDN Data Plane*

. Problem Definition: An attacker can produce a large amount of table-miss flow entries in messages to consume resources in the data plane [28, 34]. The impact of this data-to-control plane saturation attack differs for various target applications. For example, a load balancing application is more vulnerable than a hub application since it requires high programming complexity to handle complicated computations for load balancing. The controller installs flow rules in the switch flow table. The flow rules can be installed proactively or retroactively. Since the switch has a limited number of flow tables, the data plane is vulnerable to saturation attacks.

Learning Objectives: Students will learn different attack techniques to saturate data plane resources and understand the different characteristics of SDN applications that can affect flow rules proactively or reactively. Students will be able to understand the internal architecture of the data plane.

Learning Outcomes: Students understand the packet processing policies for different types of SDN applications. They can generate UDP or ICMP based flood attacks to launch saturation attacks in the data plane [28]. They can identify table-miss cases.

Challenging Questions: Students will brainstorm ways to keep the major functionalities of the SDN infrastructure working under a saturation attack in the data plane.

#### 3.1.3 **Lab 3***: API Misuse Attacks to the SDN Controller*

. Problem Definition: Networking functionalities can be implemented in software as applications on top of the control plane in SDN. Each application has its own distinct functional requirements for accessing the controller. Fallacious network applications that misuse APIs in the controller can cause serious security threats to network resources, services, and functions through the control plane due to lack of authentication and authorization for applications and lack of standard open APIs [20]. Students will explore how these unprivileged applications can crash the controller and launch memory leakage attacks [25].

Learning Objectives: Students will learn the internal structure of the controller and understand how applications can misuse APIs to cause attacks.

Learning Outcomes: Students can implement SDN applications and understand the architecture for interaction between applications and controller APIs. Students will be able to identify additional software vulnerabilities by investigating the internal code space.

Challenging Questions: Students will brainstorm security requirements for SDN applications. Different applications must have different privileges to access network information and resources. For example, a load balancing application needs network statistics while Intrusion Detection Systems (IDS) need to scrutinize packet header fields.

#### 3.1.4 **Lab 4***: Man-in-the-middle Attacks in the SDN Data Plane*

. Problem Definition: OpenFlow vulnerability is caused by insecure network management protocols where the adoption of transport layer security has lagged. Even though OpenFlow specifications support the use of Transport Layer Security (TLS), many vendors of both switches and controllers have failed to fully implement the specifications due to the complexity of TLS configuration, such as switch certificates, signing of certificates with a site-wide private key, and installing correct keys and certificates on all devices. The lack of adequate TLS implementation enables adversaries to infiltrate OpenFlow networks through a man-in-the-middle attack [6, 8]. Attackers place a device on a communication path between the switch and the controller, or simply copy the flow to his/her machine. As a result, attackers can fully control any down-stream switches and execute stealthy eavesdropping attacks in-band (i.e., links carry both data and OpenFlow traffic).

Learning Objectives: By using security protocols in the transport layer, student will learn how to establish a secure communication between the controller and switch. Students will understand OpenFlow protocol vulnerability.

Learning Outcomes: Students will be able to launch the man-in-the-middle attack in SDN and understand how attackers can steal information. They will learn security protocols like TLS, IPSec, and SSH and their usage between the controller and the switches. They will learn authentication methods needed for all devices connected to the controller or switches to ensure secure communication.

Challenging Questions: Students will brainstorm efficient authentication methods based on existing SDN features to reduce communication costs.

#### 3.1.5 **Lab 5***: Topology Poisoning Attacks to the SDN Controller*

. Problem Definition: The controller has a vast amount of important data, such as topological information, device information, and link information, all of which can be compromised by attackers. To accomplish this, attackers exploit the host tracking service in the controller. They can tamper with host location information to break through the controller and impersonate the target host. In that case, all traffic on the target host will be routed to the attacker's host. TopoGuard [14] was the first to demonstrate network poisoning attacks designed to compromise the network topology information based on the LLDP (Link Layer Discovery Protocol) protocol. It is but one example of many possible network poisoning attacks in SDN.

Learning Objectives: Students will learn link discovery methods to discover network topology. They will be able to identify important data structures in the controller, such as the topology. They can understand new attacks that compromise the controller database.

Learning Outcomes: Students will be able to generate a host location hijacking attacks by using the LLDP protocol. They can understand the procedure for maintaining network topology information in SDN. They can develop other exploits to poison other information in SDN.

Challenging Questions: Students will brainstorm a method to verify switch port properties by using model checking or symbolic execution to defend against the topology poisoning attack.

Table 1 summarizes the five security labs in SDN. The prerequisite for these labs is basic knowledge related to traditional networks and security. Without the prerequisite, it may be difficult to understand the difference between traditional network security problems and new SDN security problems. The table identifies learning outcomes, the concrete results that students will be able to accomplish after finishing each lab. After finishing all of the labs, students will understand new security issues with overall SDN knowledge.

**Table 1: Five Lab Modules for SDN Security in CloudLab**

| Lab Num. | Pre-requisite Knowledge | Learning outcomes in a new SDN Security |
|---|---|---|
| Lab 1 | Flooding attacks | Communication logic in the controller<br>Connection establishment<br>Flow rule installation |
| Lab 2 | Flooding attacks | Flow table management in the data plane |
| Lab 3 | Malicious applications<br>Software vulnerability | Core APIs in the controller<br>System architecture for SDN controller |
| Lab 4 | MITM attack<br>Security protocols like SSH | OpenFlow protocol vulnerability<br>Authentication in SDN |
| Lab 5 | LLDP | Link discovery in SDN |

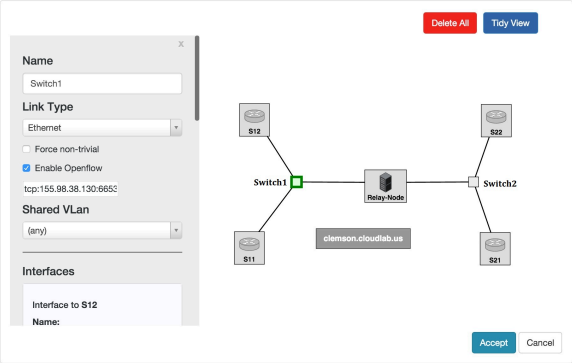



**Figure 1: Network Topology Generated in CloudLab-based Laboratory**

## 3.2 Example Lab Modules

In this section, we demonstrates two pilot lab modules for teaching students in SDN security by using CloudLab. Each lab consists of three steps: a brief introduction to CloudLab; a detailed explanation of each security problem; and the steps to launch an attack in CloudLab. The introduction to CloudLab is shared by all labs to ensure that students are familiar with the CloudLab environment. However, each lab will have different problem statements and steps to execute corresponding attack.

We next illustrate how to set up our CloudLab-based open laboratory. First, the instructor creates a profile describing the resources required for the experiment. Each experiment requires at least two OpenFlow switches and one other node (Relay-Node). The network topology is shown in Figure 1. On the left, the properties of the selected entity are listed, in this case the switch, Switch1. The instructor can check the Enable Open-flow option and assign a valid OpenFlow controller (tcp:155.98.38.130:6653, not shown in this topology), which will convert Switch1 to an OpenFlow switch. On the right, the GUI visualizes the current network topology, which is helpful to the instructor when the network topology becomes quite complicated. Once the design is accepted, a profile will automatically be generated. The instructor can then distribute this profile by providing the students with a link to it. The shared link is always to the latest version of this profile, so the instructor need not republish every time a profile is updated. Note that each profile for each lab will be somewhat different depending on attack scenarios. Students can easily copy and paste the shared link into a web browser address bar to instantiate an experiment. The procedure of instantiation is as simple as clicking the "next" button, prompting CloudLab to automatically perform the set up. Once the experiment is successfully set up, the instructor is able to see an entry for this experiment under the instructor's account. The instructor can then log in to any of the assets with root privileges if she/he wants to troubleshoot, monitor the experiment's procedure, or assess the experiment.

The SDN security problem for each lab was discussed in the previous section. Based on that information, students can understand the scope of each lab problem. Next, we use the Lab 2 and the Lab 4 as examples to articulate the detail of our lab design.

The Lab 2 addresses flooding attacks to the SDN data plane. This lab aims to design a new type of DoS attack in the SDN data plane. As mentioned before, the lab includes major components: an introduction to CloudLab; a detailed explanation of flooding attacks in the data plane; and launching a flooding attack in CloudLab. In this lab, students first create nodes shown in Figure 1 with predefined profiles for CloudLab. Second, students generate a number of Ping packets moving from one node to the other nodes and check the flow entries in the flow tables in the data plane. Flow rules are installed up to the maximum number of available flow entries. The large number of Ping packets results in the maximum being reached very quickly so that new flow rules overflow the flow entries in the data plane. Therefore, they cannot be installed and are dropped. Figure 2 shows the result of DoS attacks in the data plane. When the flow table on a switch is full, upon receiving an instruction to install a flow rule, the switch detects that there is no more space in the flow table. As the switch cannot install this rule, it sends an OFPT_ERROR message to the controller with the error code OFPFMFC_TABLE_FULL. It then drops this packet. The switch cannot forward buffered packets until there is space in the flow table to install new flow rules.

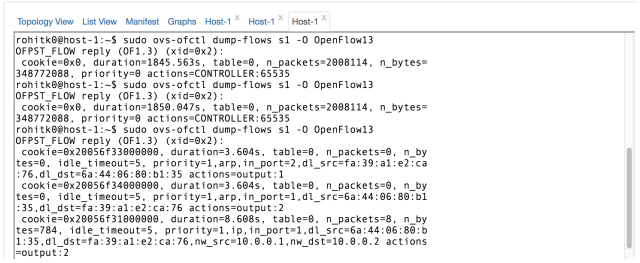



**Figure 2: A Screenshot after DoS Attacks in the SDN Data Plane**

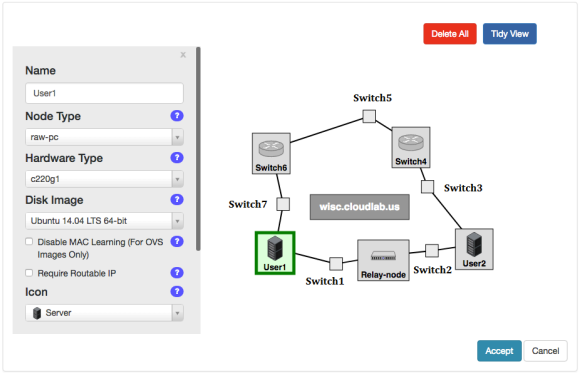
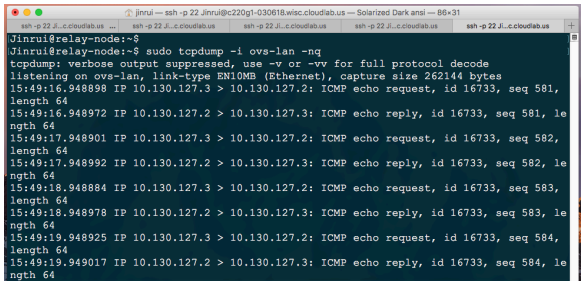



**Figure 3: Network Topology Generated in CloudLab-based Laboratory for MITM Attack Lab.**

In the Lab 4, a man-in-the-middle (MITM) attack in the SDN data plane is addressed. Students are able to launch an MITM attack in an SDN network and understand how attackers can steal information by using TLS, IPSec, and SSH protocols. The topology of the lab design has two hosts and seven switches, as shown in Figure 3. Due to space limitations, we will not explain the profile used to create the topology, but it will be shared with students in the lab. Students will set up the routing path from Host 1 to Host 2, for example with Switch 3, Switch 4, Switch 5, Switch 6, and Switch


**(a) No Traffic on Switch 4**




**(b) Sniffed Traffic on Relay Node**

**Figure 4: Screenshots after MITM Attack**

7 by installing flow rules in these OpenFlow-enabled switches, as shown in Figure 3. By using TCPdump, students can easily check the traffic passing through the designated path from Host 1 to Host 2. An attacker can cause the relay node shown in Figure 3 between Switch 1 and Switch 2 to forge a virtual link in the relay node by using the following commands:

- ovs-vsctl add-br ovs-lan
- ovs-vsctl add-port ovs-lan eth0
- ovs-vsctl add-port ovs-lan eth1
- ifconfig eth0 0
- ifconfig eth1 0
- ifconfig ovs-lan 10.130.127.1 netmask 255.255.255.0 up
- ovs-vsctl set-controller ovs-lan tcp:128.104.222.42:6653

An SDN-enabled virtual switch is created in the relay node machine and it is connected to the SDN controller. Because the SDN controller does not authenticate the SDN virtual switch, the switch can easily be registered in the SDN controller with the relay-node MAC address. Finally, the attacker captures all of the traffic between Host 1 and Host 2 through the relay node even though the hosts are able to ping each other. As shown in Figure 4, the traffic does not pass through the original path from Switch 3 to Switch 7. The traffic can only be captured between Switch 1 and Switch 2 through the relay node.

## 4 USER STUDY FOR SDN SECURITY LABS IN CLOUDLAB

We performed a user study with the sample labs described in Section 3.2. We recruited 35 students who have taken an SDN security course, which is an elective course to cover the fundamental techniques related to SDN. Thus, the 35 students are familiar with the SDN technique. In addition, the 35 students understood basic security concepts and general attacks, such as DoS attack and MITM attack, in networks. However, it was their first time to use CloudLab to study SDN security. The students conducted the sample lab modules and participated in a survey to evaluate the effectiveness

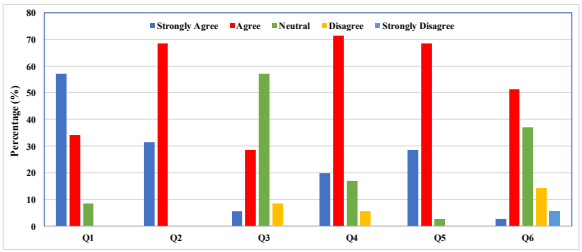



**Figure 5: Survey Results. X-axis represents each question. Y-axis indicates the percentage for each answer.**

of the designed labs. The survey asks students the following six questions:

- Q1: The lab helps students understand SDN security issues.
- Q2: The lab provides good hands-on skills to understand SDN security issues.
- Q3: The lab material is well-designed and easy to practice.
- Q4: The lab encourages students to be interested in SDN security issues.
- Q5: CloudLab helps students to understand a practical SDN environment.
- Q6: The lab makes student understand the difference between traditional network security issues and SDN specific security issues.

Figure 5 shows the results of this survey. More than 90% of students gave positive feedback on all evaluation questions except for the survey statements Q3 and Q6. Around 90% of students can understand SDN security issues even though none of the participating students has prior experience in SDN security issues. Therefore, around 90% of the students agreed that the designed lab in CloudLab would be an effective platform to understand real SDN security issues through hands-on experience. Furthermore, around 92% of the students showed research interests in SDN security after finishing the sample labs.

Even though they enjoyed the labs, some students gave negative feedback about the lab set-up and procedure. For the third survey statement (Q3), only around 35% of students can easily follow the instructions to finish the two labs. The rest of them showed a little difficulty to perform attacks in the CloudLab-based SDN environment. The reason is that it is the first time for all of them to use CloudLab and study the specific SDN security problems. They suggested putting video clips for demos since they lacked the security background to clearly understand the lab materials in SDN security. For this reason, around 57% of students gave a neutral response to the third survey statement (Q3). However, the survey showed that CloudLab is an effective platform to learn security issues. All students could finish one lab in less than five hours.

To summarize, our SDN security labs designed based on CloudLab results in positive feedback from students. The labs can provide a strong motivation for students to learn SDN security knowledge. The labs can also improve student learning skills based on CloudLab.

## 5 RELATED WORK

Many recent efforts have been devoted to addressing various security challenges in SDN, such as scanning attack prevention [15, 19],

DDoS attack detection [9], and topology poisoning attack prevention [12, 14]. Cloud-based education has become a promising trend for teaching cybersecurity [5, 21, 23, 30, 32]. Previous work has proposed hands-on cybersecurity assignments based on cloud services for cybersecurity education [5, 21, 23]. In [32], the authors focused on using the EDURange framework, a cloud-based resource for hosting on-demand interactive cybersecurity scenarios. In [32], the researchers introduced an assessment of student performance after they performed a network reconnaissance exercise. They explored the use of command line history and visualization to simplify the assessment of student performance [30]. However, none of the previous work on Cloud-based security have designed SDN security labs, despite the prevalence of SDN in networks today. This work is the first to structure hands-on security labs for SDN security education with the open cloud platform, CloudLab.

## 6 CONCLUSION

Cybersecurity education is critical for the development of future cybersecurity professionals, and for protecting IT assets. The advent of new network technology, SDN, has resulted in a high degree of flexibility in network infrastructure, but at the same time brought new security challenges. We have designed CloudLab-based SDN security labs to help students understand specific security issues in SDN and to encourage them in doing SDN security research. Our user study shows that students were satisfied with our designed SDN security labs in terms of high usability, effective cost, and hands-on experience.

## ACKNOWLEDGMENT

This work was partially supported by grants from National Science Foundation (NSF-DGE-1723663, NSF-DGE-1723804, and NSF-DGE-1723725). Dr. Younghee Park is a corresponding author.

## REFERENCES

[1] 2014. CloudLab. (2014). https://www.cloudlab.us/.
[2] 2014. RSpec. (2014). http://groups.geni.net/geni/wiki/GENIExperimenter/RSpecs.
[3] 2014. Two methods to generate a profile in RSpec in CloudLab. (2014). http://docs.cloudlab.us/index.html.
[4] 2017. geni-lib. (2017). https://bitbucket.org/barnstorm/geni-lib.
[5] Mhd Wael Bazzaza and Khaled Salah. 2015. Using the Cloud to Teach Computer Networks. In *Utility and Cloud Computing (UCC), 2015 IEEE/ACM 8th International Conference on*. IEEE, 310–314.
[6] Kevin Benton, L. Jean Camp, and Chris Small. 2013. OpenFlow Vulnerability Assessment. In *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN '13)*. ACM.
[7] Kevin Benton, L Jean Camp, and Chris Small. 2013. Openflow vulnerability assessment (Poster). In *Proceedings of ACM SIGCOMM workshop on Hot topics in software defined networking (HotSDN'13)*. ACM, 151–152.
[8] Kevin Benton, L. Jean Camp, and Chris Small. 2015. Secure communication Between OpenFlow Switch and Controllers. In *Proceedings of the Seventh International Conference on Advances in Future Internet*. AFIN.
[9] Braga Braga, Mota Mota, and Passito Passito. 2010. Lightweight DDoS flooding attack detection using NOX/OpenFlow. In *Proceedings of the 2010 IEEE 35th Conference on Local Computer Networks (LCN'10)*. IEEE, 408–415.
[10] Martin Casado, Michael J Freedman, Justin Pettit, Jianying Luo, Nick McKeown, and Scott Shenker. 2007. Ethane: Taking control of the enterprise. In *Proceedings of the ACM SIGCOMM 2007 conference*. ACM.
[11] Martin Casado, Tal Garfinkel, Aditya Akella, Michael J Freedman, Dan Boneh, Nick McKeown, and Scott Shenker. 2006. SANE: a protection architecture for enterprise networks. In *Proceedings of the 15th conference on USENIX Security Symposium*. USENIX Association.
[12] Mohan Dhawan, Rishabh Poddar, Kshiteej Mahajan, and Vijay Mann. 2015. SPHINX: Detecting Security Attacks in Software-Defined Networks. In *Proceedings of the 22nd Annual Network and Distributed System Security Symposium (NDSS'15)*.
[13] Albert Greenberg, Gisli Hjalmtysson, David A Maltz, Andy Myers, Jennifer Rexford, Geoffrey Xie, Hong Yan, Jibin Zhan, and Hui Zhang. 2005. A clean slate 4D approach to network control and management. *ACM SIGCOMM Computer Communication Review* 35, 5 (2005), 41–54.
[14] Sungmin Hong, Lei Xu, Haopei Wang, and Guofei Gu. 2015. Poisoning Network Visibility in Software-Defined Networks: New Attacks and Countermeasures. In *Proceedings of the 22nd Annual Network and Distributed System Security Symposium (NDSS'15)*.
[15] Jafar Haadi Jafarian, Ehab Al-Shaer, and Qi Duan. 2012. Openflow random host mutation: transparent moving target defense using software defined networking. In *Proceedings of ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN'12)*. ACM, 127–132.
[16] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, and Min Zhu. 2013. B4: Experience with a globally-deployed software defined WAN. In *ACM SIGCOMM Computer Communication Review*, Vol. 43. ACM, 3–14.
[17] Diego Kreutz, Fernando Ramos, and Paulo Verissimo. 2013. Towards secure and dependable software-defined networks. In *Proceedings of ACM SIGCOMM workshop on Hot topics in software defined networking (HotSDN'13)*. ACM, 55–60.
[18] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review* 38, 2 (2008), 69–74.
[19] Syed Akbar Mehdi, Junaid Khalid, and Syed Ali Khayam. 2011. Revisiting traffic anomaly detection using software defined networking. In *Proceedings of the 14th international conference on Recent Advances in Intrusion Detection (RAID'11)*. Springer-Verlag, 161–180.
[20] T. Nadeau. 2011. Software driven networks problem statement by Network Working Group. (2011). https://tools.ietf.org/html/draft-nadeau-sdn-problem-statement-00.
[21] Khaled Salah. 2014. Harnessing the cloud for teaching cybersecurity. In *Proceedings of the 45th ACM technical symposium on Computer science education*. ACM, 529–534.
[22] Khaled Salah, Mohammad Hammoud, and Sherali Zeadally. 2015. Teaching Cybersecurity using the Cloud. *IEEE Transactions on Learning Technologies* (2015).
[23] Khaled Salah, Mohammad Hammoud, and Sherali Zeadally. 2015. Teaching Cybersecurity Using the Cloud. *IEEE Transactions on Learning Technologies* 8, 4 (2015), 383–392.
[24] Seungwon Shin and Guofei Gu. 2013. Attacking Software-defined Networks: A First Feasibility Study. In *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN '13)*. ACM.
[25] Seungwon Shin, Yongjoo Song, Taekyung Lee, Sangho Lee, Jaewoong Chung, Phillip Porras, Vinod Yegneswaran, Jiseong Noh, and Brent Byunghoon Kang. 2014. Rosemary: A robust, secure, and high-performance network operating system. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 78–89.
[26] Seungwon Shin, Vinod Yegneswaran, Phillip Porras, and Guofei Gu. 2013. AVANT-GUARD: scalable and vigilant switch flow management in software-defined networks. In *Proceedings of the 20th ACM conference on Computer and communications security (CCS'13)*. ACM, 413–424.
[27] Haopei Wang, Lei Xu, and Guofei Gu. 2015. FloodGuard: A DoS Attack Prevention Extension in Software-Defined Networks. In *Dependable Systems and Networks (DSN), 2015 45th Annual IEEE/IFIP International Conference on*. IEEE, 239–250.
[28] Haopei Wang, Lei Xu, and Guofei Gu. 2015. FloodGuard: A DoS Attack Prevention Extension in Software-Defined Networks. In *Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'15)*.
[29] Richard Weiss, Michael Locasto, Jens Mache, and Vincent Nestler. 2013. Teaching cybersecurity through games: a cloud-based approach. *Journal of Computing Sciences in Colleges* 29, 1 (2013), 113–115.
[30] Richard Weiss, Michael E Locasto, and Jens Mache. 2016. A reflective approach to assessing student performance in cybersecurity exercises. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. ACM, 597–602.
[31] Richard Weiss, Jens Mache, and Michael Locasto. 2014. EDURange: hands-on cybersecurity exercises in the cloud. *Journal of Computing Sciences in Colleges* 30, 1 (2014), 178–180.
[32] Richard S Weiss, Stefan Boesen, James F Sullivan, Michael E Locasto, Jens Mache, and Erik Nilsen. 2015. Teaching cybersecurity analysis skills in the cloud. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. ACM, 332–337.
[33] Guang Yao, Jun Bi, and Luyi Guo. 2013. On the cascading failures of multi-controllers in Software Defined Networks. In *Network Protocols (ICNP), 2013 21st IEEE International Conference on*.
[34] Ying Zhang, N. Beheshti, and M. Tatipamula. 2011. On Resilience of Split-Architecture Networks. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*.