# Teaching SDN Security Using Hands-on Labs in CloudLab

Xiaohong Yuan, Zhipeng Liu
xhyuan@ncat.edu; zliu2@aggies.ncat.edu
North Carolina A&T State University

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

Hongxin Hu, Hongda Li
Clemson University

*Abstract - Software-Defined Networking (SDN) represents a major transition from traditional hardware-based networks to programmable software-based networks. While SDN brings visibility, elasticity, flexibility, and scalability, it also presents security challenges. We designed a course to introduce the emerging topics of SDN/NFV related technologies to university students. Hands-labs on SDN security on CloudLab platform were used in the course. This paper describes the hands-on SDN security labs, and our teaching experience of the course. The hands-on labs can be adopted by other instructors to teach SDN security.*

**Keywords**

***Software-defined networking (SDN, Network function virtualization (NFV), CloudLab, Security, Teaching***

## 1. INTRODUCTION

SDN offers a centralized, programmable and visible network that can dynamically evolve to the needs of businesses[1]. In comparison to a traditional network, the distinctive characteristics of a SDN include the separation of control plane and data plane, a centralized view embodied in a simplified device acting as controller, virtualizations to all functions within the

network, and the openness to change[2]. According to Google, Google fully utilized wide-area networks with SDN-based network management[3]. SDN shares close affiliations with Network Function Virtualization (NFV). NFV is the concept of offering abstractions of hardware as key network functionalities, such as firewall, network connections and load balancing[4]. Overwhelming management complexity, high costs, unscalability and slow market deployment rate are just a few notable drawbacks hardware-based network functions present[5]. Serverless, built on the basics of NFV, is a fast emerging new paradigm in virtualization and has already significantly changed the economics of offloading computations to the cloud[6].

Significant granularity, visibility, flexibility, and elasticity are definite advantages SDN and NFV bring to networking, but new security challenges are identified as well[5]. Several key security challenges in SDN have been identified and addressed, such as scanning attack prevention [7, 8], distributed denial-of-service (DDoS) attack detection [9], saturation attack mitigation [10,11], and topology poisoning attack prevention [12,13], Man-in-the-Middle (MITM) attacks[14][15].

Park, Hu, Hong, & Li (2018) identified that the use of cloud computing to be extremely effective delivery approach for cybersecurity education, but commercial cloud platforms, such as Amazon Web Services (AWS), are expensive are restrictive to certain security labs. To meet the high demands of cybersecurity educators, they proposed an open laboratory platform named CloudLab to create hands-on labs in.

CloudLab is sponsored by NSF for academic researchers to develop and experiment new cloud architectures and new cloud computing applications[5]. CloudLab is an easy-to-setup infrastructure created on the cloud for scientific research purpose on cloud computing. CloudLab is distributed infrastructure building clusters at three sites: Clemson University, University of Utah, University of Wisconsin-Madison. CloudLab combines an estimate of 5,000 cores and 500 Terabytes of storage in latest virtualization technology. For every node connecting, CloudLab provides SDN technology such as 2x10

Gbps network interfaces. A 100 Gbps full-mesh SDN interconnect lets researchers instantiate a wide range of in-cluster experimental topologies.

CloudLab supports, but not limited to, OpenFlow standard, which is an open standard protocols that organize and monitor flows. CloudLab can be easily used in two-step process: step 1 - create a user profile to encapsulate every resource component needed for the experiment (hardware, storage, network resources and software artifacts); step 2 - instantiate the created profile and instantiate the virtualized experiment within minutes, in contrast to traditional methods, thus reducing request and wait times, as well as redeployment time if a profile needed to be shared.

A distinct gap exist between explanations of emerging SDN and NFV technologies and university course curricula across the nation[5]. This course module provides an introduction to Software Defined Network (SDN)/Network Functions Virtualization (NFV), and discusses the attacks to the three main layers of SDN, and defense techniques shown in the current research. Students will complete hands-on labs that demonstrate the security issues of SDN/NFV and defense techniques.

This paper is organized as follows. Section II provides a listing of all the labs used for this course. Each listing includes the lab's description, learning objectives and learning outcomes. We explicitly explain in depth for two of the ten labs. Section III describes how the course is managed and introduces our teaching methods. Section IV describes the teaching experience accumulated for this module. Section V concludes this paper.

## 2. SDN LABS

The SDN security labs in CloudLab consist of ten lab modules. Each module has a lab description to address a specific topic in SDN, a learning objective and a list of learning outcomes. Here are labs and contents in detail. A flow is defined as a series of packets that behave in identical way.

Lab 1 Starting with CloudLab

Lab Description: Students will log on to CloudLab and create a profile. While creating the profile, students will create a topology that includes four Virtual Machines(VM). Students will create an experiment based on the profile and instantiate the topology by selecting an available cluster. After instantiation, students then check connections by sending the Ping command for all the nodes within the topology.

Learning Objectives: Upon lab completion, students should be able to create a profile, which includes a simple network topology, and start an experiment in CloudLab.

Learning Outcomes: Students will learn how to create a network topology in CloudLab, how to instantiate the topology, and confirm that the nodes in the topology are connected correctly.

Lab 2 Software Defined Networking

Lab Description: This lab introduces students to one of the popular open-source SDN controllers named Floodlight. Floodlight's source code is written with Java and deployable on any operating system.

Learning Objectives: This lab aims to let students understand the basic SDN by using Floodlight in CloudLab.

Learning Outcomes: Based on the topology that students create in the first lab (Lab 1), students can push flow rules through Floodlight controller.

Lab 3 MITM Attack with Flow Rule Manipulation

Lab Description: The controller is responsible for flow settings in switches such that all flow processing in the data-path is based on instructions from the controller. The controller then sets the flow rules in switch flow tables to either forward the flow packets to a particular port or drop packets coming from that particular source. The flow rules change depending on different network topologies, various user requests, and network protocols. This lab demonstrates how an user/attacker can modify flow rules using static flow pusher.

Learning Objectives: To educate students how to conduct a MITM attack through simple flow rule manipulation through the Representational State Transfer (REST) API.

Learning Outcomes: Students will gain firsthand experience in flow rule manipulation. Students will also learn how to utilize some features on the controller using REST API.

Lab 4 Flooding Attacks to the SDN Data Plane

Lab Description: The saturation attack from data plane to control plane differs from various attack applications. A load balancing application is vulnerable since it needs to handle high amount of complicated computations. The controller installs flow rules in the switch flow table. Attackers can produce a large amount of table-miss flow entries in messages to consume resources in the data plane. This action will force controller to install reactive flow rules since no matches will be found in flow table. Since the switch has a limited number of flow tables, the data plane is vulnerable to saturation attacks.

Learning Objectives: This lab teaches how to conduct a saturation attack, understand the different characteristics of SDN applications. This lab also teaches the internal architecture of the data plane.

Learning Outcomes: Students will learn packet processing policies for SDN applications. Students can launch UDP or ICMP based flood attacks to launch saturation attacks in the data plane.

Lab 5 Man-in-the-middle Attacks in the SDN Data Plane

Lab Description: The lack of security implementation for the Transport Layer Security (TLS) enables adversaries to infiltrate OpenFlow networks through a MIMT. Attackers can simply place a device on a communication path between the switch and the controller. As a result, attackers can fully control any down-stream switches and execute stealthy eavesdropping attacks (listen on any OpenFlow communications on this flow path).

Learning Objectives: Students will understand OpenFlow protocol vulnerability, thus in turn understand the importance of establishing a secure communication between controller and switch.

Learning Outcomes: Students will be able to launch the MIMT in SDN and understand how attackers can steal information. Students can further learn security protocols like TLS, IPSec, and SSH and potentially deploy these protocols between the controller and the switches.

Lab 6 API Misuse Attacks to the SDN Controller

Lab Description: SDN is dependent on the normal operations of NorthBound API and SouthBound API. While SouthBound API involves communicating protocol standards to establish handshakes between devices, NorthBound API is expandable. Through both sets of APIs, 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 or malicious 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.

Learning Objectives: Students will learn the internal structure of the controller and understand how applications can misuse APIs to cause attacks. Students will explore how these unprivileged applications can crash the controller and launch memory leakage attacks.

Learning Outcomes: Students can learn the interactions between applications and controller APIs through hands-on experience.

Lab 7 Local Host Hijacking

Lab Description: The controller is of the utmost importance in a SDN. Information includes transmission data, topological understanding, device data, and link data, all can be sensitive information that cannot fall in the wrong hands. However, vulnerabilities exist that allows attackers to exploit on. One way for attacker is to 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 route to the attacker's host. This lab demonstrates network poisoning attacks designed to compromise the network topology information based on the LLDP (Link Layer Discovery Protocol) protocol.

Learning Objectives: This lab aims to educate students what a network poisoning attack is and how the vulnerability is being taken advantage of.

Learning Outcomes: Students will learn how to simulate a vulnerable network that's susceptible to network/topology poisoning attacks. Students will learn how to conduct a network/topology poisoning attack.

Lab 8 Segregating Flows

Lab Description: The switch acts as first line of filter for flows (a series of packets behaving the same way) in the data plane of SDN before the flows are allowed to be forwarded to the controller in the control plane. However, if conflicting flows occur, the switch may not be able to function normally, and parties involve in the conflicting flows may not receive their packets as expected. This lab demonstrates how to segregate flows.

Learning Objectives: This lab introduces what qualifies as a conflicting flow rule.

Learning Outcomes: Students will learn how to segregate flows to ensure data packets are received for intended users.

Lab 9 FlowVisor

Lab Description: FlowVisor is a special purpose OpenFlow controller that acts as a transparent proxy between OpenFlow switches and multiple OpenFlow controllers. FlowVisor creates rich slices of network resources and delegates control of each slice to a different controller. Slices can be defined by any combination of switch ports (layer 1), src/ dst ethernet address or type (layer 2), src/dst IP address or type (layer 3), and src/dst TCP/ UDP port or ICMP code/type (layer 4). FlowVisor enforces isolation between each slice, i.e., one slice cannot control another's traffic.

Learning Objectives: To educate students to learn to write flow rules in FlowVisor. A flow rule is named as a slice by FlowVisor designer.

Learning Outcomes: Students will learn how to you will learn how to slice OpenFlow network and have each slice controlled by a separate controller. In the process, you will also learn the concept of flowspaces and how the centralized visibility and "layerless-ness" of OpenFlow enables flexible slicing.

Lab 10 Resolve Conflicting Flows

Lab Description: The switch acts as first line of filter for flows (a series of packets behaving the same way) in the data plane of SDN before the flows are allowed to be forwarded to the controller in the control plane. However, if conflicting flows occur frequently and switch is unable to respond, the flows are forwarded to controller and remain idle for the duration of the connection, this may lead to potential serious DoS attacks. This lab demonstrates how to resolve such conflicts with priority approach.

Learning Objectives: This lab introduces what qualifies as a conflicting flow rule and the common OpenFlow parameters.

Learning Outcomes: Students will learn how to identify conflicting rules. Students will also learn how to use priority approach to resolve flow conflicts.

## 3. COURSE ON SDN SECURITY

This course module was taught in a special topic graduate level course titled Advanced Security for Emerging Nets at North Carolina A&T State University in Spring 2019 semester. This course meets face-to-face twice a week. Fifteen students enrolled in the class.

Upon completion of this course, we expect students to be able to:

- Explain the key components of SDN/NFV architecture and concepts
- Explain the major security issues in different layers of SDN/NFV
- Identify defense techniques for attacks to SDN/NFV
- Conduct research, and give presentations/tutorials on their research
- Conduct implementation-oriented hands-on labs related to SDN/NFV security

Almost every week of the semester, the students were asked to complete one of the ten listed labs. The students were then graded on completion of the lab. Each student submitted their work in the form of either screenshots of steps or video recordings. As for the final project of the course, each student was asked to submit a SDN/NFV related topic to research and develop a new lab on. Each student were to present their project in an approximate 10-minute span. The final projects were assessed on their presentation skills, knowledge base, critical thinking and overall impressions. Then students were then given a lab survey and a course survey. The results of labs and surveys are discussed in the next section.

This course was designed in seminar style, executed through guided inquiry collaborative learning[20][21]. Each student was assigned to prepare materials to either teach one to several chapters of selected textbook[22] or teach and demo a lab. This style requires students to study, prepare and have adequate subject knowledge in this subject, in return the students enhance their teaching skills while stimulating other students to actively participate in discussions, and promote thinking[16][17]. Students demonstrated creativity and utilized many teaching methods and tools, including Plickers, Kahoot, YouTube videos and Powerpoint slides. Past researches indicated that use of gamification tools have significant addition to project-based learning[18]. One student even taught the class using similar method to POGIL teaching. The student created the teaching material as handouts. Students first had to read to build up

knowledge, then discussed in groups before finally answering the assessment questions from the handouts. The mixing of these teaching methods increase learner motivation, enhance review of technical content and bring an upbeat atmosphere. Previous research reflects the use of gamification tools allow faculty to clearly identify whether the students have successfully mastered the concepts and allow instructors to further structure peer-to-peer active learning more effectively in class[19].

## 4. TEACHING EXPERIENCE

An anonymous student survey was conducted on the course module. This section presents the results from student survey.

A total of twelve students participated in the survey. Students' self-ranking on knowledge attained in learning objectives for the labs show that eighty-three point four percent (83.4%) strongly agreed or agreed that the learning objectives of the labs are met.

Even though eighty-three percent (83%) of students believe labs are somewhat difficult, seventy-five percent (75%) of students believe that they are more interested in computer security after taking this course. Seventy-five percent (75%) students expressed having either high or very high interests in the labs. Majority of students also commented they wish to apply the knowledge learned in this course to their own research areas. One hundred percent (100%) of students recognized SDN and NFV as easy to deploy and advantageous to any other methods they've experienced using.

## 5. CONCLUSION

This paper describes a course module designed to teach students about SDN security knowledge through hands-on labs in CloudLab, and how the SDN related security vulnerabilities can be exploited. The course module consists of ten hands-on lab exercises simulating various attacks as well as delivering core foundation knowledge. Students were also asked to create new labs that were

fully tested. Student were required to teach the course through guided inquiry collaborative learning under the supervision of the professor.

The course module was taught in Spring 2019 semester. Our teaching experience proved that students felt interested and learned effectively. This course module may be adopted by instructors teaching network security, web security, and network functions.

Since students from the current course designed new labs for the subject, these labs may be included as part of the course in the future. More sophisticated labs can also be introduced for the course. Potential subjects that can be taught in the course are serverless[6], lightweight virtualization[28], and IoT management[29].

## 6. ACKNOWLEDGEMENTS

This work was partially supported by grants from National Science Foundation (NSF-DGE-1723663, NSF-DGE-1723804, and NSF-DGE-1723725).

## REFERENCES

1. Kreutz, D., Ramos, F. M., Verissimo, P., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 103(1), 14-76.
2. Goransson, P., Black, C., & Culver, T. (2016). Software defined networks: a comprehensive approach. Morgan Kaufmann.
3. Vahdat, A., Clark, D., & Rexford, J. (2015). A purpose-built global network: Google's move to SDN. Queue, 13(8), 100.
4. "Network Function Virtualisation - Introductory White Paper," https://portal.etsi.org/nfv/nfv white paper. pdf.
5. Park, Y., Hu, H., Yuan, X., & Li, H. (2018, February). Enhancing Security Education Through Designing SDN Security Labs in CloudLab. In Proceedings of the 49th ACM Technical Symposium on Computer Science Education (pp. 185-190). ACM.

6. Aditya, P., Akkus, I. E., Beck, A., Chen, R., Hilt, V., Rimac, I., ... & Stein, M. (2019). Will Serverless Computing Revolutionize NFV?. Proceedings of the IEEE.

7. Giotis, K., Argyropoulos, C., Androulidakis, G., Kalogeras, D., & Maglaris, V. (2014). Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments. Computer Networks, 62, 122-136.

8. S. A. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting traffic anomaly detection using software defined net- working," in Proceedings of the 14th international conference on Recent Advances in Intrusion Detection (RAID'11). Springer-Verlag, 2011, pp. 161–180.

9. B. Braga, M. Mota, and P. Passito, "Lightweight ddos flooding attack detection using nox/openflow," in Proceedings of the 2010 IEEE 35th Conference on Local Computer Networks (LCN'10). IEEE, 2010, pp. 408–415.

10. Lim, S., Yang, S., Kim, Y., Yang, S., & Kim, H. (2015). Controller scheduling for continued SDN operation under DDoS attacks. Electronics Letters, 51(16), 1259-1261.

11. Mohammadi, R., Javidan, R., & Conti, M. (2017). Slicots: An sdn-based lightweight countermeasure for tcp syn flooding attacks. IEEE Transactions on Network and Service Management, 14(2), 487-497.

12. M. Dhawan, R. Poddar, K. Mahajan, and V. Mann, "Sphinx: Detecting security attacks in software-defined networks," in Proceedings of the 22nd Annual Network and Distributed System Security Symposium (NDSS'15), February 2015.

13. S. Hong, L. Xu, H. Wang, and G. Gu, "Poisoning network visibility in software-defined networks: New attacks and countermeasures," in Proceedings of the 22nd Annual Network and Distributed System Se- curity Symposium (NDSS'15), February 2015.

14. Yao, J., Han, Z., Sohail, M., & Wang, L. (2019). A Robust Security Architecture for SDN-Based 5G Networks. Future Internet, 11(4), 85.

15. Mabel, J. P., Vani, K. A., & Babu, K. R. M. (2019). SDN Security: Challenges and Solutions. In Emerging Research in Electronics, Computer Science and Technology (pp. 837-848). Springer, Singapore.

16. Spruijt, A., Jaarsma, A. D. C., Wolfhagen, H. A. P., van Beukelen, P., & Scherpbier, A. J. J. A. (2012). Students' perceptions of aspects affecting seminar learning. Medical teacher, 34(2), e129-e135.

17. Spruijt, A., Wolfhagen, I., Bok, H., Schuurmans, E., Scherpbier, A., Van Beukelen, P., & Jaarsma, D. (2013). Teachers' perceptions of aspects affecting seminar learning: a qualitative study. BMC medical education, 13(1), 22.

18. Khan, A., Ahmad, F. H., & Malik, M. M. (2017). Use of digital game based learning and gamification in secondary school science: The effect on student engagement, learning and gender difference. Education and Information Technologies, 22(6), 2767-2804.

19. Leung, E., & Pluskwik, E. (2018). Effectiveness of Gamification Activities in a Project-based Learning Classroom.

20. Hanson, D. M. (2006). Instructor's guide to process-oriented guided-inquiry learning. Lisle, IL: Pacific Crest.

21. Shih, J. L., Chuang, C. W., & Hwang, G. J. (2010). An inquiry-based mobile learning approach to enhancing social science learning effectiveness. Journal of Educational Technology & Society, 13(4), 50-62.

22. Goransson, P., Black, C., & Culver, T. (2016). Software defined networks: a comprehensive approach. Morgan Kaufmann.

23. H.Wang, L. Xu, and G. Gu, "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) , June 2015.

24. Fruhlinger, J. (2018). The Mirai botnet explained: How teen scammers and CCTV cameras almost brought down the Internet. CSOonline, March, 9.

25. Oh, B. H., Vural, S., Wang, N., & Tafazolli, R. (2018). Priority-Based Flow Control for Dynamic and Reliable Flow Management in SDN. IEEE Transactions on Network and Service Management, 15(4), 1720-1732.

26. Zhang, M., Bi, J., Bai, J., Dong, Z., Li, Y., & Li, Z. (2017, August). FTGuard: A Priority-Aware Strategy Against the Flow Table Overflow Attack in SDN. In SIGCOMM Posters and Demos (pp. 141-143).

27. Izard, R. (2019). Project Floodlight. Retrieved from https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/pages/1343518/Static+Entry+Pusher+API [Accessed 29 Apr. 2019].

28. Morabito, R. (2019). Lightweight Virtualization in Edge Computing for Internet of Things.

29. Zarca, A. M., Bernabe, J. B., Trapero, R., Rivera, D., Villalobos, J., Skarmeta, A., ... & Gouvas, P. (2019). Security Management Architecture for NFV/SDN-aware IoT Systems. IEEE Internet of Things Journal.