Diversity Modeling to Evaluate Security of Multiple SDN Controllers

Hellen Maziku
Tennessee State University
Nashville, TN

Sachin Shetty
Old Dominion University
Norfolk, VA

Dong Jin
Illinois Institute of Technology
Chicago, IL

Charles Kamhoua
Army Research Lab
Adelphi, MD

Laurent Njilla and Kevin Kwiat
Air Force Research Lab
Rome, NY

***Abstract*—Software Defined Networking (SDN) facilitates network orchestration and ability to reconfigure the network plane at run time. Despite ubiquitous adoption of SDN, there is also growing concern about security risks posed by SDN. The security risks stem from centralized controller, trust issues between network elements due to Open APIs and an insecure OpenFlow channel. However, SDN controllers are a common target since compromising SDN controllers has the capability of impairing the entire network. SDN controller protection efforts have centered around replicating controllers for redundancy and hardening host Operating Systems. There is lack of research efforts on the security impact of adopting multiple SDN controllers. In this paper, we present a network diversity modeling framework to assess impact on security risk due to multiple SDN controllers. Using attack graphs and diversity models, we explore the security impact of resource relationships to SDN multiple controller networks. Our results reveal that having similar resource instances in different multiple SDN controllers increases the security risk.**

***Index Terms*—Cyber Resilience; Security Metrics; SDN security; Diversity Modeling; OpenFlow; Attack graphs**

I. Introduction

Software defined networking (SDN) is a networking paradigm to provide automated network management at run time through network orchestration and virtualization. SDN is used primarily for quality of service (QoS) and automated response to network failures. SDN allows decoupling of the control and data plane, enabling logically centralized network controllers to manage whole networks [11]. Current critical infrastructures were designed with a static non adaptive nature which makes it practically infeasible to reconfigure a network to react to cyber attacks. Our previous work in [10], [8], and [9], demonstrates how SDN's dynamic and real-time reconfigurability ability is the answer to cyber security and resilience of todays critical infrastructures such as smart grids and cloud networks.

However, there is a growing concern about security risks in adopting SDN. SDN presents new security challenges due to centralized control logic that maybe prone to DoS attacks (single point of failure), trust issues between network elements due to Open APIs and an OpenFlow channel that may not be secure, depending on the configuration options enforced [14], [1]. There have been efforts proposed to secure SDN across all layers [5], [14], [4]. Specific approaches, such as, authentication mechanisms such as TLS, shared secret passwords and nonces to avoid eavesdropping and spoofed southbound communications have been proposed. However, the SDN controllers are a fairly common target because impairing the controllers can severely compromise large network segments. Security mitigation schemes to protect SDN controllers include replicating controllers for redundancy and hardening host Operating Systems. However, there needs to be a systematic understanding of the degree of security enhancements multiple SDN controllers can provide. Some of the key questions that need to be addressed include; Does overall security of SDN improve with multiple controllers? What is the optimal number of SDN controllers to provide the desired degree of protection? What are the cost implications of choosing a High Availability (HA) SDN controller configuration?

This paper provides a first step towards formally modeling SDN controller diversity. Using diversity modeling principles in [17], we investigate the hypothesis that adding multiple controllers improves the security of an SDN enabled network. We evaluate existing diversity metrics and analyze the applicability of diversity modeling to SDN multiple controller frameworks. We propose a framework with multiple controllers as depicted in Figure 1 for improving security and resilience of SDN enabled infrastructures. We adopt a high Availability (HA) hierarchical role-based controller architecture in the SDN control layer. Each controller is assigned specific roles by another controller that acts as master and delegator. The challenge is to come up with a cost model to determine optimal number of controllers. We use single-controller and three-controller networks to model diversity and our results demonstrate that adding multiple controllers improves the security of an SDN enabled network. We employ attack graphs to model the casual relationships between different resources running in the SDN network and use diversity models to evaluate the security impact of resource relationships to SDN multiple controller networks. We reveal that having similar resource instances in different SDN elements in the network lowers network diversity and permits reuse of exploits by attackers.

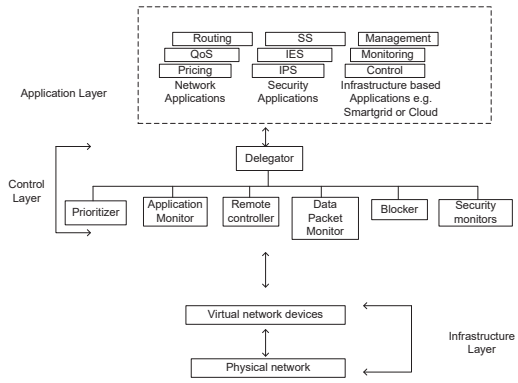


Fig. 1. A Framework with Hierarchical multiple Controllers for SDN enabled networks

II. Related work

A. SDN vulnerability assessment

Since its discovery, SDN has gained increasing acceptance, mainly due to the unlimited possibilities of attack mitigation strategies present through SDN adoption. On the down-side, SDN comes with its own security challenges most of which have been documented in existing literature [1], [14], [6]. The authors in [1] examine the vulnerabilities in SDN's most popular flavor, OpenFlow and classify them in three main groups; lack of TLS adoption, centralization of the control plane and untrusted northbound open APIs. Focusing on lack of TLS adoption, according to [1], one out of eight controller implementations and OpenFlow switch vendors supports TLS. The rest of the switch and controller vendors lack motivation for TLS hence opt for a plain-text TCP control channel opening doors to man-in-the-middle attacks. Centralizing all control operations introduces single point of failure. Flooding large volumes of message flows such as *Packet-In messages* or *Flow-Mod messages* would DoS a controller, or switch. Using a set of crafted packets and arp poisoning, authors in [3] and [12] exploit SDN vulnerabilities in Open Floodlight networks. Researchers have proposed security mitigation measures to protect SDN, which include; cryptography, enforcing SDN trust policies, replication and diversity of SDN components such as controllers, protocols or software images [7], [1], [15]. Recent SDN controller platforms such as ONOS and OpenDaylight are distributed in nature and hence support clustering of controllers for scalability and reliability. There is still no existing work to formally prove that high availability controller architectures improve the security of SDN networks.

B. Network Diversity Modeling as a Security Metric

A non-diverse network is vulnerable to an attack that exploits a single weakness that is recurring in all its components. Existing diversity implementations have focused on software based approaches such as instruction set randomization, address and data space randomization [2] and topology-aware software assignment for enhancing the robustness of network routing [13]. Wang et al. in [17] introduce diversity modeling as a global property of an entire network for the purpose of evaluating robustness of the network against zero day attacks. Drawing analogy from biodiversity in ecology, the authors in [17] propose three security metrics for modeling network diversity.

Diversity metric 1: Borrowing concepts from familiar mathematical models of biodiversity in ecology such as *species richness* and *Shannon-Wiener index*, Wang et al. propose the first diversity metric based on distinct number of resources in a network. Given a network G with a total number of hosts $H = \{h_1, h_2, ..., h_n\}$ and a set of resource types $R = \{r_1, r_2, ..., r_m\}$ with the resource mapping $res(.)$. Let the number of resource instances be given as $t = \sum_{i=1}^{n} |res(h_i)|$ and relative frequency of each resource be given as $p_i = \frac{|\{h_i : r_j \in res(h_i)\}|}{t} (1 \leq i \leq n, 1 \leq j \leq m)$. $r(G)$ known as networks effective richness of resources is given as: $r(G) = \prod_1^n P_i^{-p_i}$. Network diversity based on effective richness is defined as d_1 in equation below. A higher value of d_1 represents a more diverse network.

$$d_1 = \frac{r(G)}{t} \tag{1}$$

Diversity metric 2: The second diversity metric is derived from an attack graph of a network and reflects how attackers may compromise a critical asset, also known as a goal condition in a network, with the least effort. We model an attack graph which is syntactically equivalent to a resource graph in [17], but models known SDN vulnerabilities rather than zero day attacks. Given an attack graph $G(E \bigcup C, R_r \bigcup R_i)$ with pre and post condition relations R_r, R_i and a goal condition $c_g \in C$, for each $c \in C$ and $q \in seq(c)$ where $seq(c)$ is a set of attack paths $\{e_1, e_2, ..., e_n : \langle e_n, c \rangle \in R_i\}$ for a given sequence of exploits $e_1, e_2, ..., e_n$, denote $R(q)$ for $\{r : r \in R, r \text{ appears in } q\}$. Diversity based on least attacking effort is a ratio between minimum number of distinct resources on a path and minimum number of steps on a path. Network diversity based on least attacking effort is defined below as d_2. This ratio can never exceed 1.

$$d_2 = \frac{min_{q \in seq(c_g)} |R(q)|}{min_{q' \in seq(c_g)} |q'|} \tag{2}$$

Diversity metric 3: The least attacking effort also known as the shortest path to the attacker's target does not provide a full picture of the threat and hence carries insufficient information [17]. The third metric, with the help of probability, combines all paths in an attack graph and gives the average attacking effort. Assume p is the probability of achieving the final goal condition in a network where all resources are different (no exploit reuse), and $p^{'}$ is the probability of achieving the final goal condition in the same network but with the possibility of reusing an exploit. p and $p^{'}$ represent the attack likelihood with respect to the attacker's goal condition and both probabilities are modeled using a Bayesian network derived from the attack graph. Network diversity based on average attacking effort is defined as:

$$d_3 = \frac{p}{p^{'}} \tag{3}$$

III. Implementation of SDN Controller Diversity

This section uses two SDN controller configuration examples and diversity metrics from the previous section to evaluate and quantify the security in SDN multiple controller networks.

A. Single SDN Controller Configuration

Figure 2 represents a single OpenFlow controller network with three open vSwitches (X, Y, Z) and three hosts (A, B, C). The attacker is on host A and aims to attack switch Z or host C using two threat vectors eavesdropping and DoS. Suppose the controller is running firewall, REST API and Load Balancer services. The services running in the controller plus the OpenFlow instance trigger data plane flows such as: *Packet_in*, *Flow_mod*, *Features_request*, *Feautures_reply*, *arp* in the network, giving 5 total resource instances.

Effective Richness of Resources: Using equation 1, diversity based on the effective richness of resources in the SDN single controller network, d_1 is 3.789.

Least Attacking Effort: In order to compute network diversity based on the least attacking effort, we build an attack graph to model control plane vulnerabilities in our single controller network as depicted in Figure 3 (ignore probability values inside and outside the rectangles). A pair represents a security based condition (e.g., connectivity (source, destination) or privilege (privilege, host)). The triple tuple depicts potential exploit of resource, (resource, exploiting host, exploitable host). Edges flow from pre-conditions to exploits (e.g., from (A, X) and (user, A) to (arp, A, X)), and from that exploit to its post-conditions (e.g., from (arp, A, X) to (user, Y)). We observe five attack paths as illustrated by Table I. Using equation 2, diversity based on the least attacking effort in the SDN single controller network gives a ratio $d_2 = \frac{\min(1,2,3)}{\min(1,2,3)} = 1$. This ratio indicates that the current network is not diverse and there is 100% potential improvement in diversity.

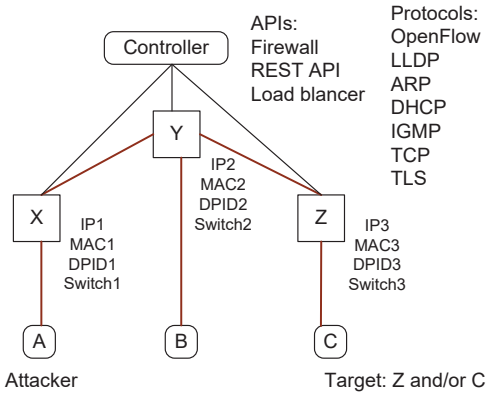


Fig. 2. Single SDN Controller Network Configuration

Average Attacking Effort: In order to compute the cumulative probability of successfully executing an exploit, in this case, the exploit is <*features_reply, A, Z*>, we update the attack graph in Figure 3 to include individual and cumulative probability scores for conditions and exploits. Given exploit e, condition c and probabilities for individual scores $p(e)$ and $p(c)$, cumulative scores $P(e)$ and $P(c)$ can be obtained using equations:

–$P(e) = p(e) \cdot \prod_{c \in R_r(e)} P(c)$ and

–$P(c) = p(c)$ if $R_i(c) = \varnothing$ otherwise $P(c) = p(c) \cdot \bigoplus_{e \in R(c)} P(e)$ for any $e \in E$ and $\bigoplus(S_1 \cup S_2) = \bigoplus S_1 + \bigoplus S_2 - \bigoplus S_1 \cdot \bigoplus S_2$ for any disjoint and non-empty sets $S_1 \subseteq E$ and $S_2 \subseteq E$ [16].

Cumulative scores in an attack graph factor in the casual relationships between exploits and conditions. This cumulative score exposes the difference in attack likelihood between two multiple SDN controller networks with same number of controllers but different configurations such as different topology setups, or different applications/software running within the controllers. For individual scores (probabilities inside the rectangles), we convert NVD and CVSS base scores [16] for SDN vulnerabilities. We use the above cumulative probability scores equations to obtain cumulative scores for the exploits and conditions in the one controller attack graph (probabilities outside the rectangles). The Conditional Probability Tables (CPT) in a Bayesian network help to calculate the joint probability function for achieving a certain goal. For example, in the single controller network configuration, Table II helps to calculate the probability of exploiting the <*features_reply, A, Z*> resource at switch Z. As seen in Figure 3, 0.264 represents the cumulative probability score for achieving the final goal condition. This probability for exploiting the network includes the significance of causal relationships among resources running in the different controllers, therefore factoring in the effect of how the controllers are positioned in the network.

TABLE I
Attack Paths for Single SDN Controller Network

Attack Path	Steps	resources
<features_reply, A, Co>	1	1
<arp, A, X> → <packet_in, Y, Co>	2	2
<arp, A, X> → <flow_mod, Co, Y>	2	2
<arp, A, X> → <packet_in, Y, Co> → <features_reply, A, Z>	3	3
<arp, A, X> → <flow_mod, Co, Y> → <features_reply, A, Z>	3	3

B. Three SDN Controller Configuration

Figure 4 represents a second degree SDN multiple controller network with six open vSwitches ($X1$, $Y1$, $Z1$,$X2$, $Y2$, $Z2$,) and three OpenFlow controllers ($C1$, $C2$, $C3$). An attacker at $X1$ aims to attack switch $Z2$. Controllers $C2$ and $C3$ are running control plane firewalls while $C1$ runs REST API and Load Balancer applications. Similar to single controller network, the services running in the controllers plus the OpenFlow instance trigger data plane flows such as: *Packet_in* at C1, C2, C3, X2, Z2, *Features* at C3, X2, *arp* at Z1, Z2, *firewall* at C2, C3 in the network, giving 11 total resource instances.

Effective Richness of Resources: Using equation 1, di-

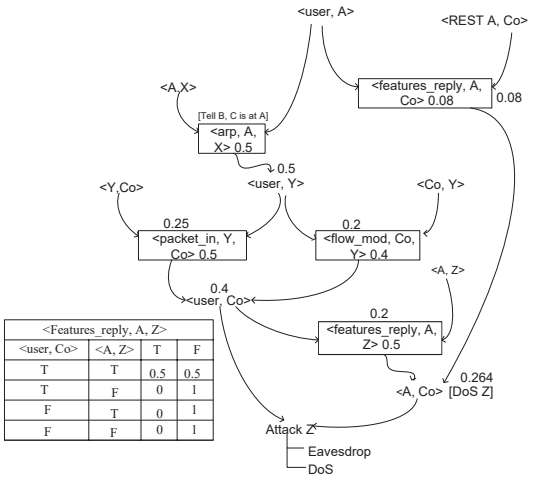


Fig. 3. Modeling network diversity of a single controller network using Bayesian networks

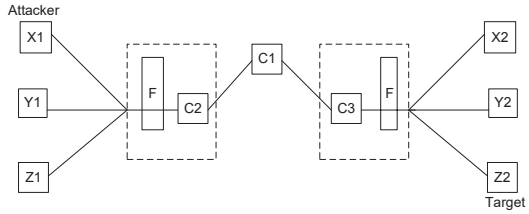


Fig. 4. A Three SDN Controller Network Example

versity based on the effective richness of resources in the SDN three controller network is d_1 is 3.902, showing an improvement in diversity compared to the diversity score of single controller network (3.789).

Least Attacking Effort: Figure 5 is an attack graph that models control plane vulnerabilities for the three controller network in Figure 4. From the attack graph in Figure 5 (Ignore probability values inside and outside the rectangles), there are 24 attack paths available for the attacker at $X1$ to attack switch $Z2$.Table II gives the number of resources and attack path steps needed to reach the final goal. The Table only shows 12 out of 24 attack paths since these carry sufficient information to illustrate diversity, the rest are obtained in a similar manner. Using equation 2 and Table II, diversity based on the Least Attacking Effort gives a ratio of $d_2 = \frac{\min(3,4)}{\min(5,6,7)} = 0.6$. This ratio indicates that there is 60% potential improvement in diversity (present diversity is 40%). This demonstrates the improvement in diversity as we move from a configuration of one controller to a configuration of three SDN controllers.

Average Attacking Effort: We update the attack graph in Figure 5 to include individual and cumulative probability scores for conditions and exploits. The cumulative probability score for achieving the final goal condition, attacking switch Z in a three-controller network configuration becomes 0.0001255, which is an improvement compared to the cumulative probability score of 0.264 from the single controller network.

TABLE II
ATTACK PATHS FOR THREE SDN CONTROLLER NETWORK

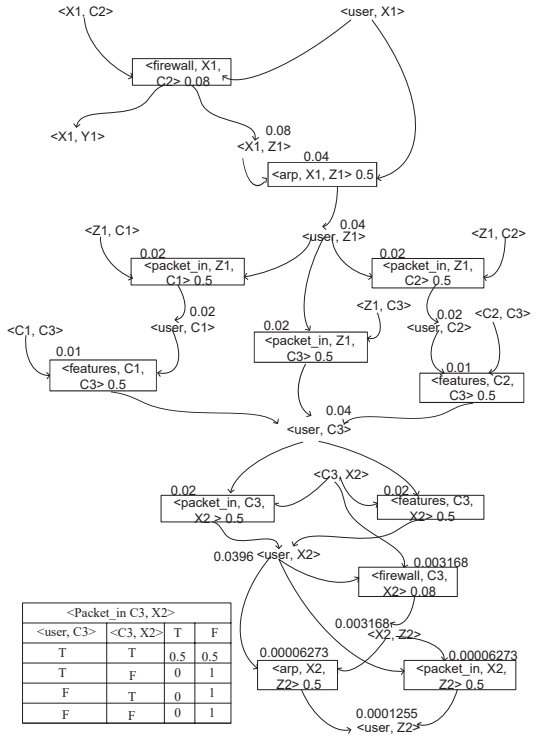
Attack Path	Steps	Resources
<firewall, X1, C2> --> <arp, X1, Z1> --> <packet_in, Z1, C1> --> <features, C1, C3> --> <packet_in, C3, X2> --> <arp, X2, Z2>	6	4
<firewall, X1, C2> --> <arp, X1, Z1> --> <packet_in, Z1, C1> --> <features, C1, C3> --> <packet_in, C3, X2> --> < packet_in, X2, Z2>	6	4
<firewall, X1, C2> --> <arp, X1, Z1> --> <packet_in, Z1, C1> --> <features, C1, C3> --> <packet_in, C3, X2> --> <firewall, C3, X2> --> <arp, X2, Z2>	7	4
<firewall, X1, C2> --> <arp, X1, Z1> --> <packet_in, Z1, C1> --> <features, C1, C3> --> <packet_in, C3, X2> --> <firewall, C3, X2> --> < packet_in, X2, Z2>	7	4
<firewall, X1, C2> --> <arp, X1, Z1> --> <packet_in, Z1, C3> --> <packet_in, C3, X2> --> <arp, X2, Z2>	5	3
<firewall, X1, C2> --> <arp, X1, Z1> --> <packet_in, Z1, C3> --> <packet_in, C3, X2> --> < packet_in, X2, Z2>	5	3
<firewall, X1, C2> --> <arp, X1, Z1> --> <packet_in, Z1, C3> --> <packet_in, C3, X2> --> <firewall, C3, X2> --> <arp, X2, Z2>	6	3
<firewall, X1, C2> --> <arp, X1, Z1> --> <packet_in, Z1, C3> --> <packet_in, C3, X2> --> <firewall, C3, X2> --> < packet_in, X2, Z2>	6	3
<firewall, X1, C2> --> <arp, X1, Z1> --> <packet_in, Z1, C3> --> <features, C3, X2> --> <arp, X2, Z2>	5	4
<firewall, X1, C2> --> <arp, X1, Z1> --> <packet_in, Z1, C3> --> <features, C3, X2> --> < packet_in, X2, Z2>	5	4
<firewall, X1, C2> --> <arp, X1, Z1> --> <packet_in, Z1, C3> --> <features, C3, X2> --> <firewall, C3, X2> --> <arp, X2, Z2>	6	4
<firewall, X1, C2> --> <arp, X1, Z1> --> <packet_in, Z1, C3> --> <features, C3, X2> --> <firewall, C3, X2> --> < packet_in, X2, Z2>	6	4

network.

With a probabilistic approach, diversity in a network refers to the probability, where if an attacker can successfully achieve a certain goal condition in the network, he/she can still achieve the targeted goal even if all of network's resources were to be different across all components. Therefore we model the attack likelihood while considering the effect of reusing an exploit on different network components. Consider Figure 5, assume that reusing the *packet_in* exploit on controller $C3$ increases the probability from 0.5 to 0.9. Figure 6 shows the updated Bayesian network with the effect of reusing the exploit. Diversity based on the average attacking effort as discussed in equation 5 is a ratio between probability of achieving final goal condition with no exploit reuse and probability of achieving the same goal condition with exploit reuse. Looking at Figure 5 and Figure 6, $d_3 = \frac{0.0001255}{0.0002444} = 0.514$. We observe that modeling diversity on the three-controller network using the least attacking effort gives a higher diversity value (60%) but masks the effect of re-using an exploit. The average attacking effort however, gives a lower metric value (51%) but exposes the effect of having different resource instances in the network (all resources appearing only once) as opposed to reusing exploits.

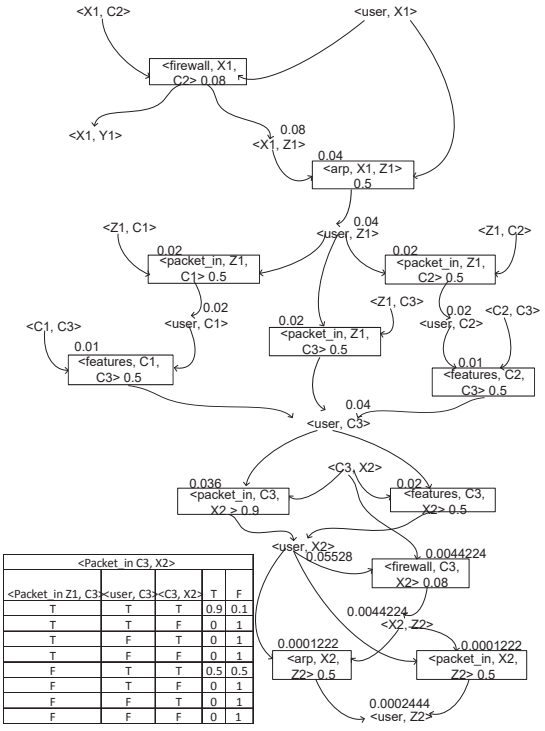
IV. CONCLUSION AND FUTURE WORK

This paper lays a foundation on methods of formally quantifying the security of SDN multiple controller networks. We extend existing network diversity modeling principles to Software Defined Networking. Using diversity modeling, we demonstrate that adding multiple controllers improves the security of an SDN enabled network. We use diversity models and attack graphs to evaluate the security impact of resource relationships to SDN multiple controller networks.



<Packet_in C3, X2>			
<user, C3>	<C3, X2>	T	F
T	T	0.5	0.5
T	F	0	1
F	T	0	1
F	F	0	1

Fig. 5. A Bayesian network for the three-controller network configuration



<Packet_in C3, X2>				
<Packet_in Z1, C3>	<user, C3>	<C3, X2>	T	F
T	T	T	0.9	0.1
T	T	F	0	1
T	F	T	0	1
T	F	F	0	1
F	T	T	0.5	0.5
F	T	F	0	1
F	F	T	0	1
F	F	F	0	1

Fig. 6. Modeling network diversity of a three-controller network using Bayesian networks: Effect of reusing resources

Our preliminary results reveal that having similar resource instances in different SDN elements in the network lowers network diversity. We are currently extending the probabilistic diversity metric to factor in higher degree SDN multiple controller networks and the cost of diversity. We are looking at how we can improve existing metrics to factor in degree of importance of each controller.

ACKNOWLEDGMENT

This material is based upon work supported by the Department of Energy under Award Number DE-OE0000780 and Office of the Assistant Secretary of Defense for Research and Engineering agreement FA8750-15-2-0120.

REFERENCES

[1] Kevin Benton, L Jean Camp, and Chris Small. Openflow vulnerability assessment. In *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, pages 151–152. ACM, 2013.

[2] Sandeep Bhatkar and R Sekar. Data space randomization. In *DIMVA*, pages 1–22. Springer, 2008.

[3] Jeremy M Dover. A denial of service attack against the open floodlight sdn controller, 2013.

[4] Raphael Durner and Wolfgang Kellerer. The cost of security in the sdn control plane. *CoNEXT Student Workhop*, 2015.

[5] Scott Hogg. Sdn security attack vectors and sdn hardening. [Online].

[6] Sungmin Hong, Lei Xu, Haopei Wang, and Guofei Gu. Poisoning network visibility in software-defined networks: New attacks and countermeasures. In *NDSS*, 2015.

[7] Diego Kreutz, Fernando Ramos, and Paulo Verissimo. Towards secure and dependable software-defined networks. In *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, pages 55–60. ACM, 2013.

[8] Hellen Maziku and Sachin Shetty. Network aware vm migration in cloud data centers. In *Research and Educational Experiment Workshop (GREE), 2014 Third GENI*, pages 25–28. IEEE, 2014.

[9] Hellen Maziku and Sachin Shetty. Towards a network aware vm migration: Evaluating the cost of vm migration in cloud data centers. In *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on*, pages 114–119. IEEE, 2014.

[10] Hellen Maziku and Sachin Shetty. Software defined networking enabled resilience for iec 61850-based substation communication systems. In *Computing, Networking and Communications (ICNC), 2017 International Conference on*, pages 690–694. IEEE, 2017.

[11] Nick McKeown. Software-defined networking. *INFOCOM keynote talk*, 17(2):30–32, 2009.

[12] Yoav Francis Nir Solomon and Liahav Eitan. Floodlight openflow ddos. https://www.slideshare.net/YoavFrancis/floodlight-openflow-ddos. [Online].

[13] Adam J O'Donnell and Harish Sethu. On achieving software diversity for improved network security using distributed coloring algorithms. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 121–131. ACM, 2004.

[14] Sandra Scott-Hayward, Sriram Natarajan, and Sakir Sezer. A survey of security in software defined networks. *IEEE Communications Surveys & Tutorials*, 18(1):623–654, 2016.

[15] Amin Tootoonchian and Yashar Ganjali. Hyperflow: A distributed control plane for openflow. In *Proceedings of the 2010 internet network management conference on Research on enterprise networking*, pages 3–3, 2010.

[16] Lingyu Wang, Tania Islam, Tao Long, Anoop Singhal, and Sushil Jajodia. An attack graph-based probabilistic security metric. *Lecture Notes in Computer Science*, 5094:283–296, 2008.

[17] Lingyu Wang, Mengyuan Zhang, Sushil Jajodia, Anoop Singhal, and Massimiliano Albanese. Modeling network diversity for evaluating the robustness of networks against zero-day attacks. In *European Symposium on Research in Computer Security*, pages 494–511. Springer, 2014.