Blessing or Curse? Revisiting Security Aspects of Software-Defined Networking

Lisa Schehlmann, Sebastian Abt and Harald Baier
da/sec – Biometrics and Internet Security Research Group
Hochschule Darmstadt, Darmstadt, Germany
{Lisa.Schehlmann,Sebastian.Abt,Harald.Baier}@h-da.de

***Abstract*—Software-Defined Networking (SDN) is an emerging technology, physically separating data and control planes of network devices. From a security point of view SDN has two sides. First, it enables network security functions *by design*, because traffic flows can be redirected or filtered based on packet content or application layer state – functionality, which to date requires additional network security devices like firewalls, intrusion detection systems or spam filters in conventional networks. On the other hand, due to physical separation of planes, SDN possibly offers additional attack vectors compared to traditional network architectures, which may severely impact overall network availability as well as confidentiality, authenticity, integrity and consistency of network traffic and control data. In this paper, we discuss and balance security provided by SDN with security threats of SDN also in respect of traditional networks. We develop an evaluation methodology for both sides and show that from a security point of view SDN is a blessing for today's and future network design and operation.**

***Keywords*—Software-Defined Networking, OpenFlow, network security**

I. Introduction

Software-Defined Networking (SDN) [1] is a new network paradigm which physically separates the control and data planes of a network device. Specifically, the control plane is removed from a network device and implemented on a specialized central controller. As a result, the single controlling instance maintains a global view on the network and serves as dedicated point of management for the whole network. The data plane resides on the device and forwards network traffic based on remotely programmed forwarding rules. Forwarding rules can be defined by specific applications running on top of the controller and can be based on various input information, e.g., specifics of packet headers transmitted in a network or application layer information. Consequently, SDN provides an application programming interface (API) allowing a network's data plane to be altered by external applications.

This new concept is two-sided with respect to security, because it enables both new security mechanisms and new threats. First SDN provides network security functions *by design*. For instance well-known network security concepts have been transferred to SDN in order to achieve network security by SDN. In [2], [3], [4], the authors introduce FleXam, an SDN application to access packet level information by the controller. FleXam provides an easy way to integrate packet filters or firewalls into an SDN network. Furthermore [5], [6] describe FRESCO, a framework for enabling security controls in the communication between the controller and the data plane to implement network security applications.

On the other hand, due to physical separation of planes, SDN offers additional attack vectors compared to traditional network architectures, which may severely impact overall network availability as well as confidentiality, authenticity, integrity and consistency of network traffic and control data. For example [7] gives an overview of the vulnerabilities caused by separating the control and the data plane and highlights the widespread failure to adapt the Transport Layer Security (TLS) protocol to the communication channel between switches and controllers. [8] shows threat vectors of SDN and proposes a design for secure and dependable SDN.

However, the community misses an assessment if the security benefits provided by SDN exceed the additional threats. That is, if SDN is a blessing or a curse with respect to network security. In this paper we develop a methodology to evaluate both security aspects of SDN. In a first step, we assess the information security threats *of* SDN by analyzing the SDN reference architecture according to fundamental security goals of information security: confidentiality, authenticity, integrity, availability and consistency. We then review the security benefits provided *by* SDN by discussing approaches, which improve network security (e.g. by developing controller applications and frameworks). In a final step we compare both aspects and conclude that the security benefits of SDN outweigh the threats, i.e., SDN is a blessing with respect to security.

The remainder of the paper is organized as follows: Sect. II provides fundamentals of SDN. We then discuss and evaluate security threats of the SDN reference architecture in Sect. III. Next, Sect. IV assesses SDN extensions that aim at providing network intrinsic security. We balance the outcome of prior sections in Sect. V in order to assess whether SDN is blessing or curse. In Sect. VI we present related work of SDN security. Finally Sect. VII summarizes and concludes the paper.

II. Background

The architectural design of SDN separates the control and data planes of a network device to provide a global view of the network and centralized control of network devices. The paradigm evolves from previous research projects: (1) The 4D-project [9], which introduces an architecture to separate the decision logic of the network from the protocols governing interaction of network elements. (2) [10] proposes a *Secure*

This work has been partly funded by the Hesse government under grant number 306/11-51 (NetFlowBot) and supported by CASED.

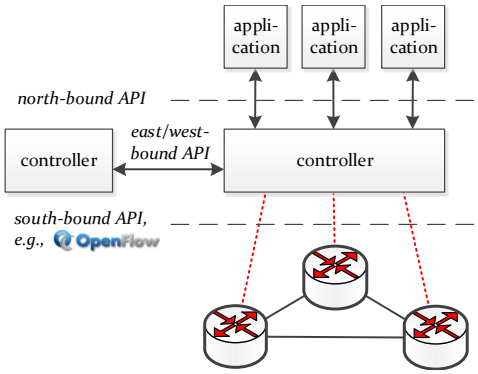


Fig. 1. Conceptual architecture of Software-Defined Networking

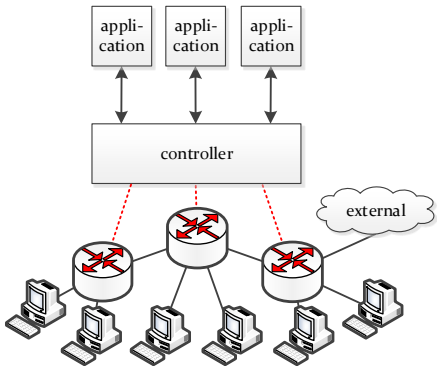


Fig. 2. Sample SDN network topology

Architecture for the Networked Enterprise (SANE) by providing a single protection layer between the Ethernet and Internet Protocol layer. (3) Ethane [11] was proposed as improved successor of SANE. Its architecture consists of a controller and Ethane switches. Ethane allows to define a single network-wide policy, which determines the fate of all network packets. The work of Ethane results in the OpenFlow protocol [12], an approach to standardize the communication between the control and data planes as well as an implementation of the first OpenFlow controller NOX [13].

The conceptual architecture of SDN is depicted in Fig. 1. The data plane still resides on the device and is responsible to process (e.g. forward, drop) network packets based on defined flow rules. The control plane is decoupled from the network device and resides at a centralized controller, which decides about the forwarding behavior of network packets. Some use cases require more than one controller, e.g., for replication purposes or large networks. Several controllers coordinate their tasks or exchange information via an east/west-bound API. The data plane communicates via the south-bound API with the control plane. The most popular south-bound API is the above mentioned OpenFlow protocol, which is standardized by the Open Networking Foundation (ONF) [14]. The controller provides a global view of the network via a north-bound API to the applications (e.g. SDN management applications). In contrast to the south-bound API, there is no standardized north-bound API. However, the ONF initiates a Northbound Interface Working Group.

III. Information Security of SDN

The SDN community already focuses on security issues of the SDN design (see Sect. VI). Some of the issues occur in conventional computer networks as well and best practices are already developed and applied. However, due to the centralized architecture of SDN (compared to the commonly decentralized architecture of conventional networks) additional issues arise.

Our methodology assesses security issues of both SDN and conventional networks regarding key aspects of information security: confidentiality, authenticity, integrity, availability and consistency. We rate the evaluation criteria by the respective impact to the network. Our evaluation scheme makes use of the impact levels $\oplus$ (uncritical), $\bigcirc$ (neutral) and $\ominus$ (critical). For better comparison, we match points to each impact level (1 pt, 0 pt, -1 pt). Subsequently, we compare the results.

TABLE I. Evaluation of Security of SDN against security of conventional networks

criterion	SDN	conventional
Confidentiality		
encryption of communication channels	$\bigcirc$ (0pt)	$\bigcirc$ (0pt)
access control for management interfaces	$\bigcirc$ (0pt)	$\bigcirc$ (0pt)
Authenticity		
network devices (e.g. routers, switches)	$\bigcirc$ (0pt)	$\bigcirc$ (0pt)
controller	$\ominus$ (-1pt)	$\oplus$ (1pt)
applications	$\ominus$ (-1pt)	$\oplus$ (1pt)
Integrity		
forwarding tables, network state	$\ominus$ (-1pt)	$\ominus$ (-1pt)
Availability		
network devices (e.g. routers, switches)	$\bigcirc$ (0pt)	$\bigcirc$ (0pt)
controller	$\ominus$ (-1pt)	$\oplus$ (1pt)
Consistency		
forwarding tables	$\ominus$ (-1pt)	$\ominus$ (-1pt)
	-5pt	1pt

$\oplus$ uncritical (1pt), $\bigcirc$ neutral (0pt), $\ominus$ critical (-1pt)

In our evaluation we assume a sample OpenFlow-based SDN topology, comprising several hosts, switches and one controller (see Fig. 2). The result of our evaluation is summarized in Table I.

A. Confidentiality

Confidentiality prevents disclosure of information to unauthorized entities. To ensure confidentiality, two common methods *encryption* and *access control* are used. (1) Encryption of the communication channel between data plane and controller means that an attacker has access to the ciphertext, but he is not able to recover the corresponding plaintext. For instance an encrypted channel may be established by TLS as proposed by the OpenFlow switch specification [15]. (2) Granting access after authentication via management interfaces of network devices and the controller means that only authorized entities have access to data structures. Access control may be enforced by the operating system.

We evaluate the impact of confidentiality to SDN networks as well as to conventional networks as neutral. Several techniques to encrypt network communication channels and apply access control are already developed and could be adapted to both network architectures.

B. Authenticity

Authenticity describes the property that entities are actually the one they claim to be. A well-known cryptographic method to ensure authenticity is a signature (e.g. a message authentication code (MAC) for bulk data). Furthermore network devices as well as the controller have to exchange keys (either secret ones for generating / validating a MAC or public keys for asymmetric signatures) [15]. To ensure trust between the applications and the controller, [8] proposes the use of an autonomic trust management system (e.g. [16]) to prevent that malicious applications bind themselves to the controller to perform malicious actions.

We evaluate the issue of authenticity for network devices in SDN networks as well as in conventional networks as neutral, because techniques for mutual authentication are already deployed. We evaluate authenticity of centralized controller and applications in SDN networks as critical, because a malicious controller or application could compromise the behavior of the whole network. Due to the lack of the controller and applications in conventional networks, we evaluate this as uncritical.

C. Integrity

Integrity means that information is unmodified during its life-cycle. In SDN networks, primarily the integrity of flow rules and messages transferred between the layers has to be ensured. Integrity of messages could also be implemented by e.g., a message authentication code (MAC).

We evaluate the issue of integrity regarding flow/forwarding rules as critical in an SDN network as well as in conventional networks, because modified rules could lead to undesirable effects.

D. Availability

Availability is the property to access data, devices and services every time when it is needed. The obvious bottleneck in our sample OpenFlow network is the controller. If the controller is unavailable due to misconfiguration, a technical error or an attack (e.g. denial of service (DoS) attack), the network devices are only able to enforce predefined rules. If an SDN switch is down due to technical errors or a DoS attack, the controller could dynamically reprogram the network paths.

Possible solutions to mitigate a DoS attack are implementing a rate limiting, reducing the timeout of flow table entries or dropping packets of a DoS attack (see [17]). [18] discusses the placement and number of controllers to achieve redundancy.

We rate the issue of availability of network devices as neutral, due to the possibility to easily change paths in SDN as well as in conventional networks. However, we evaluate the non-availability of the controller in an SDN network as critical. Due to the lack of a controller in a conventional network, this issue is evaluated as uncritical.

E. Consistency

If different applications are used to define flow rules, it is possible that the flow rules are not self-consistent. Hence a mediator between applications and controller is needed to deal with conflicting rules. One implementation to detect and mediate rule conflicts is FortNOX [19].

We evaluate the impact of conflicting rules as critical for conventional as well as for SDN networks, because it could lead to unpredictable network behavior in both architectures.

TABLE II. EVALUATION OF SECURITY BY SDN AGAINST SECURITY BY CONVENTIONAL NETWORKS

criterion	SDN	conventional
Network management		
simplicity of management	⊕ (1pt)	⊖ (-1pt)
integration of legacy/new security appliances	⊕ (1pt)	⊖ (-1pt)
centralized view	⊕ (1pt)	⊖ (-1pt)
Costs		
costs and time for error handling/maintenance	⊕ (1pt)	⊖ (-1pt)
robustness against outage	⊖ (-1pt)	⊕ (1pt)
Attack detection and mitigation		
detection/mitigation methods by design	⊕ (1pt)	⊖ (-1pt)
time to react to network attacks	⊕ (1pt)	⊖ (-1pt)
	5pt	-5pt

⊕ positive (1pt), ○ neutral (0pt), ⊖ negative (-1pt)

IV. INFORMATION SECURITY PROVIDED BY SDN

A lot of research to provide network intrinsic security by an SDN network is recently done (see Sect. VI). In this section, we assess network characteristics of SDN as well as conventional networks, which affect the security provided by a network as ⊕ (positive), ○ (neutral) or ⊖ (negative). Additionally, we match points again (1 pt, 0 pt, -1 pt), for better comparison. The result of our evaluation is summarized in Table II.

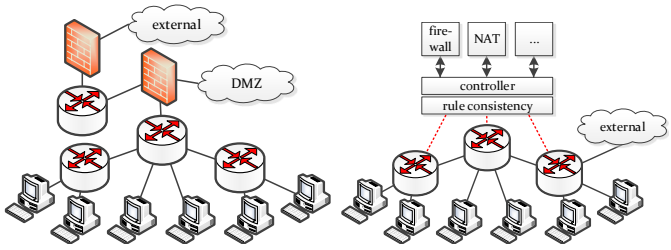
A. Network management

In contrast to conventional networks an SDN offers a centralized view. Thus, the management of an SDN is simpler than the management of conventional networks. Maintenance is more flexible due to automation, which saves costs and time for error handling and simplifies the deployment of network wide policies.

The integration of new security applications (e.g. firewall) is easier in an SDN network due to the flexibility and the global view. The integration of legacy applications is easily possible, too, but it has to be ensured, that no inconsistency is introduced.

In addition, we emphasize that SDN provides, through its global view on the network and its flexibility with respect to maintenance and reconfiguration, a natural security environment for security challenges like Bring Your Own Device (BYOD). The advantage of using SDN for the BYOD challenge is to enhance an existing SDN enabled network by additional features to deal with the problem of unmanaged devices. A sample implementation is described in [20], where the Ballarat Grammar School applied the HPs Sentinel Security app [21] in a hybrid SDN network to tackle the BYOD challenge.

We evaluate the network management of SDNs as positive, because it is more flexible than in conventional networks due



(a) Conventional network (b) SDN network

Fig. 3. Conventional vs. SDN

to the global view. Corresponding to that, we rate network management of conventional networks as negative.

B. Costs

An easy maintenance of SDN networks has the side effect of saving costs. On the one hand personnel costs, because error handling is easier and configuration effort is decreased. Furthermore, deploying SDN networks could reduce power consumption of hardware devices. On the one hand switches are no longer responsible for powerful computation tasks and on the other hand if hardware devices could be reduced, the power consumption diminishes, too. Due to saving costs for maintenance we evaluate this criterion for SDN as positive. On the contrary, we evaluate the criterion for conventional networks as negative due to the complexity of management.

An important drawback of our sample SDN network topology is the single point of failure if the controller is not available. Because of this, we evaluate robustness as negative for our sample network. One approach to tackle this disadvantage is to introduce redundancy by multiple controllers. Conventional networks are more robust against outage due to their decentralization, which is why we evaluate robustness of conventional networks as positive.

C. Attack detection and mitigation

SDN is still under development and not widely deployed, which allows network operators to integrate network attack detection and mitigation still by design. Due to this, we evaluate the possibility to integrate attack detection/mitigation to SDN by design as positive. We rate this criterion for conventional networks as negative because applying changes to conventional networks are more complex. One problem could be the lack of an agreement on a management protocol, by which solutions could not adapt to each implementation of SDN.

The most prominent network threat defense mechanisms are cryptography, firewalls, network based intrusion detection systems (NIDS), and a unified threat management (UTM). As we consider SDN as transport medium for network traffic, SDN cannot provide cryptographic measures to counter network-based sniffing or spoofing attacks. However, the remaining defense provision concepts may easily be ported to SDN as we discuss in what follows.

Fig. 3 shows on the left side a sample conventional network security topology at the perimeter including two firewalls. Each firewall only has access to the data, which flows through it. From a conceptual point of view the firewall is an additional item in the network introduced due to the legacy security lack of conventional network devices. Additionally there is no default communication channel between the different firewalls. With regard to SDN a firewall as a legacy hardware box is replaced by a software application on top of the controller as visualized on the right in Fig. 3. Then the application has access to data transferred through *every* switch in the SDN. Thus the firewall application gains a global view on the network. Additionally maintenance of the application itself with respect to software updates or reconfiguration is much easier and the number of devices decreases, which results in lower costs. Furthermore in contrast to its classical relative the firewall application does not need an additional interface to gather data as the network packets for an inspection are already provided by SDN. And besides detection or maintenance, mitigation is much easier as the firewall application enforces rules to capture packets at any device in the network in real time. Due to this, network operators will be able to react faster to attacks in SDN networks than in conventional networks, which is why we rate the time to react in SDN networks as positive and in conventional networks as negative.

A similar argumentation holds for NIDS to inspect network activity and to launch reaction or more generally for every legacy security application. Even more attractive the hosting of different security applications centrally on top of the SDN controller yields a straightforward concept to realize a Unified Threat Management System. Additionally SDN may be used to hide the actual network topology through Network Address Translation (NAT) or to provide monitoring services of a network.

V. Discussion: Blessing or Curse for Network Security?

Networks are vulnerable due to protocol flaws by design. This concerns all layers of the network stack (e.g. ordinary IP, TCP, or HTTP) and arises due to the fact that not only attackers find new ways in exploiting vulnerabilities not anticipated in protocol design, but also due to protocols not being strictly implemented and used as initially standardized.

Conventional networks do not provide any detection or mitigation measure by design. They are decentralized and thus complex and difficult to manage. However, a decentralized network architecture is robust against outages of single components. In contrast, SDN is a new paradigm to design and manage networks. The concept differs mainly from conventional networks in centralization of the control plane. While this centralization provides benefits, such as an end-to-end view on the whole network topology, it also raises new challenges, such as to ensure consistency amongst systems and availability of the central controller. From that, the question arises whether the benefits of this new paradigm may outweigh possible disadvantages, i.e. if SDN is blessing or curse. In this section we compare the (additional) security risks *of* SDN as described in Sect. III with the security benefits provided *by* SDN as described in Sect. IV.

We start with the benefits provided by SDN. The global view and control of all network devices, which is intrinsically provided by the centralized position of the controller in the

network, is the key advantage of SDN compared to decentralized networks. To improve and ensure the security of the network we can simply implement well-known, standardized, and proven security concepts (e.g. firewalls, spam filters, NIDS, or other monitoring services) as applications on top of the controller. As a proof of concept, the community already came up with a controller application to access packet level information [2], [3], [4] or an application to detect DDoS flooding attacks [22]. Compared to a bunch of legacy security appliances that would have to be deployed in conventional networks to achieve similar functionality, integration of security features in SDN seems less complex and due to large-scale view on network events we see the potential to better classify events and especially to reduce false-positive rates, a practically critical performance metric. Furthermore, due to this benefit in SDN, we also see the advantage of being able to easily share relevant information across security applications of different type (e.g. information on spam-sending hosts, as detected by spam filters, being shared with botnet detection systems). When discussing these benefits, we would like to stress that SDN security applications use the already available infrastructure of SDN to collect network packets and to subsequently mitigate detected anomalies by reprogramming the network. That is, SDN as-is offers great interfaces for network threat intelligence and reaction, notwithstanding the possibility to outsource security services by, e.g., integrating dedicated security devices. If this is preferred, network operators simply have to reconfigure the SDN to redirect certain traffic to a dedicated security box. Therefore every existing security service may be used in an SDN, too.

As security risks of SDN, we especially recognize security threats (see Sect. III) to confidentiality, authenticity, integrity, availability and consistency. Yet, we would like to remind that these threats exist in conventional networks as well. Especially, most threats are well studied in conventional networks and may be encountered by standardized protocols (e.g. TLS) or network design guidelines. In fact, we are convinced that, conceptually, SDN does not introduce any entirely new security threats due to its different design that our community has not seen before and that cannot be solved by applying already existing techniques. Specifically, in our opinion, future research should exactly focus on solving these problems of security of SDN by applying well known concepts, such as sensible integration of public key infrastructures in order to achieve encryption of communication channels and authenticity of SDN components, in order to facilitate network operators' in building secure SDN.

Comparing the results of the evaluation, we believe that security benefits provided by SDN (5pt) outweigh the drawbacks of SDN's decentralized design and associated security risks (-5pt). The resilience achieved through decentralized design of conventional networks (1pt) do not outweigh the drawback in the evaluation of security provided by conventional network architectures (-5pt). To sum up, we are convinced that SDN will be a blessing for the security of future networks if basic measures against threats to confidentiality, authenticity, integrity, availability and consistency are integrated into SDN. Security of SDN is still work in progress, but the benefits of security by SDN outweigh the security issues of SDN we encounter so far.

VI. Related work

The authors of [23] provide a survey of SDN security research. They distinguish the research into two categories: (1) The challenges of the SDN framework's architecture and (2) the security enhancements derived by SDN. They link the issues of SDN to the affected SDN layers (application, control and data layer) and the interfaces between these layers. Additionally, they categorize related work in SDN security research into security analysis, enhancements and solutions to issues mapped to the above mentioned layers and interfaces. They point out that one important issue of SDN is trust between all involved layers and an increased potential of denial of service (DoS) attacks due to SDN's centralization and limited space in flow-tables. The authors differentiate the common efforts to enhance network security by SDN, into middle-boxes (e.g. IDS) integrated into the networks and monitoring systems developed for the SDN network infrastructure.

Further research focuses either on the issues of SDN's design, solutions to tackle these issues or on network security enhancements derived by SDN. Several research performs security analyses of SDN, especially OpenFlow. [7] gives an overview of the vulnerabilities caused by separating the control and the data plane. [8] shows threat vectors of SDN and proposes a design for secure and dependable SDN. A formal security analysis of SDN architectures is provided by the Internet Draft SDN Security Requirements [24]. [17], [25] perform a security analysis of OpenFlow and propose prevention and mitigation techniques. [26] mentions challenges by implementing SDN with focus on network performance, scalability, security and interoperability and proposes possible solutions. To solve and mitigate issues of SDN's design, the data plane extension AVANT-GUARD [27] reduces data-to-control plane communication to mitigate DDoS attacks. [28] presents the permission system PermOF, which aims to give OpenFlow applications minimum permissions to prevent abuse. [2], [3], [4] introduce FleXam, an extension to access packet level information by the controller for e.g., traffic classification. [29] introduces the framework CloudWatcher to monitor large and dynamic cloud networks. [22] shows a method to detect a DDoS attack based on flow features collected in an SDN network. [30] demonstrates a method for an attacker to fingerprint an SDN in preparation to a DoS attack. [31] introduces the model checking system Flover, which verifies that OpenFlow rules do not violate the security policy of the network. [5] and [6] introduce a Framework for Enabling Security Controls in OpenFlow networks (FRESCO) to integrate security related applications to SDN. This work resulted in the SE-Floodlight controller [32], [33].

VII. Summary and Conclusion

In this paper, we revisited security aspects of SDN. Especially, we aimed at systematically comparing security benefits provided by SDN with security risks of SDN and used conventional networks as benchmark. As such, this paper extended an SDN security survey performed in [23] with a formal evaluation. From that evaluation we conclude that SDN is, security-wise, a blessing for future network design. The paradigm change promoted by SDN introduces additional attack vectors in comparison to conventional networks, but also offers great opportunities to implement network intrinsic

security. Yet, the major threats to confidentiality, authenticity, integrity, availability and consistency we identified in this paper are not new to our community and can be solved by applying already established concepts in a sensible manner. Hence, we propose that future work should focus on exactly that part. Especially, we see a demand in the integration of public key infrastructures into SDN in order to protect communication between different components of an SDN and to assure authenticity of components. The latter especially becomes an important requirement if we consider future inter-provider SDN setups and related security challenges.

REFERENCES

[1] Open Networking Foundation, "Software-Defined Networking: The New Norm for Networks," Open Networking Foundation, ONF White Paper, April 2012.

[2] S. Shirali-Shahreza and Y. Ganjali, "Empowering Software Defined Network controller with packet-level information," in *Proceedings of the 2013 IEEE International Conference on Communications Workshops*, ser. ICC '13. IEEE Computer Society, June 2013, pp. 1335–1339.

[3] S. Shirali-Shahreza and G. Yashar, "Efficient Implementation of Security Applications in OpenFlow Controller with FleXam," in *Proceedings of the 21st Annual Symposium on High-Performance Interconnects*, ser. HOTI '13. IEEE Computer Society, August 2013, pp. 49 – 54.

[4] S. Shirali-Shahreza and Y. Ganjali, "FleXam: flexible sampling extension for monitoring and security applications in openflow," in *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '13. ACM, August 2013, pp. 167–168.

[5] S. Shin, P. Porras, V. Yegneswaran, M. Fong, G. Gu, and M. Tyson, "FRESCO: Modular Composable Security Services for Software-Defined Networks," in *Proceedings of the 20th Annual Network and Distributed System Security Symposium*, ser. NDSS '13. The Internet Society, February 2013.

[6] S. Shin, P. Porras, V. Yegneswaran, and G. Gu, "A Framework For Integrating Security Services into Software-Defined Networks," in *Proceedings of the 2013 Open Networking Summit (Research Track poster paper)*, ser. ONS '13, April 2013.

[7] K. Benton, L. J. Camp, and C. Small, "OpenFlow Vulnerability Assessment," in *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '13. ACM, August 2013, pp. 151–152.

[8] D. Kreutz, F. M. Ramos, and P. Verissimo, "Towards Secure and Dependable Software-Defined Networks," in *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '13. ACM, August 2013, pp. 55–60.

[9] A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang, "A Clean Slate 4D Approach to Network Control and Management," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 5, pp. 41–54, October 2005.

[10] M. Casado, T. Garfinkel, A. Akella, M. J. Freedman, D. Boneh, N. McKeown, and S. Shenker, "SANE: A Protection Architecture for Enterprise Networks," in *Proceedings of the 15th Conference on USENIX Security Symposium - Volume 15*, ser. USENIX-SS'06. USENIX Association, July 2006, article No. 10.

[11] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "Ethane: Taking Control of the Enterprise," in *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '07. ACM, August 2007, pp. 1–12.

[12] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, March 2008.

[13] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "NOX: Towards an Operating System for Networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 3, pp. 105–110, July 2008.

[14] Open Networking Foundation, [Online] URL: http://www.opennetworking.org.

[15] ——, "OpenFlow Switch Specification," Open Networking Foundation, ONF Specification Version 1.4.0, October 2013.

[16] Z. Yan and C. Prehofer, "Autonomic Trust Management for a Component-Based Software System," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 6, pp. 810–823, Nov 2011.

[17] R. Klöti, V. Kotronis, and P. Smith, "OpenFlow: A Security Analysis," in *Proceedings of the 8th Workshop on Secure Network Protocols (NPSec), part of IEEE ICNP*, ser. NPSec '13. IEEE, October 2013.

[18] B. Heller, R. Sherwood, and N. McKeown, "The Controller Placement Problem," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, ser. HotSDN '12. ACM, August 2012, pp. 7–12.

[19] P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, and G. Gu, "A Security Enforcement Kernel for OpenFlow Networks," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, ser. HotSDN '12. ACM, August 2012, pp. 121–126.

[20] Open Networking Foundation, "OpenFlow Gives Malware a Caning," Open Networking Foundation, Customer Case Study, 2013.

[21] Hewlett-Packard Development Company, "Realizing the power of SDN with HP Virtual Application Networks," Technical White Paper, October 2012.

[22] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS Flooding Attack Detection Using NOX/OpenFlow," in *Proceedings of the 2010 IEEE 35th Conference on Local Computer Networks*, ser. LCN '10. IEEE Computer Society, October 2010, pp. 408–415.

[23] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "SDN Security: A Survey," in *2013 IEEE SDN for Future Networks and Services*, ser. SDN4FNS'13. IEEE Computer Society, November 2013, pp. 1–7.

[24] S. Hartman, M. Wasserman, and D. Zhang, "Security Requirements in the Software Defined Networking Model," Internet Engineering Task Force, Internet-Draft draft-hartman-sdnsec-requirements-01, April 2013, work in progress.

[25] R. Klöti, "OpenFlow: A Security Analysis," Master's thesis, ETH Zrich, October 2012.

[26] S. Sezer, S. Scott-Hayward, and C. Pushpinder Kaur, "Are we ready for SDN? Implementation challenges for software-defined networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 36–43, July 2013.

[27] S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "AVANT-GUARD: Scalable and Vigilant Switch Flow Management in Software-defined Networks," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, ser. CCS '13. ACM, November 2013, pp. 413–424.

[28] X. Wen, Y. Chen, C. Hu, C. Shi, and Y. Wang, "Towards a Secure Controller Platform for OpenFlow Applications," in *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '13. ACM, August 2013, pp. 171–172.

[29] S. Shin and G. Gu, "CloudWatcher: Network Security Monitoring Using OpenFlow in Dynamic Cloud Networks (or: How to Provide Security Monitoring As a Service in Clouds?)," in *Proceedings of the 2012 20th IEEE International Conference on Network Protocols (ICNP)*, ser. ICNP '12. IEEE Computer Society, October 2012, pp. 1–6.

[30] ——, "Attacking Software-defined Networks: A First Feasibility Study," in *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '13. ACM, August 2013, pp. 165–166.

[31] S. Son, S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "Model Checking Invariant Security Properties in OpenFlow," in *Proceedings of IEEE International Conference on Communications*, ser. ICC '13. IEEE Computer Society, June 2013, pp. 1974–1979.

[32] OpenFlowSec.Org, "SDN Security Suite," 2013, [Online] URL: http://www.openflowsec.org/SDNSuite.html.

[33] P. Porras, "Toward a More Secure SDN Control Layer SRI Internationals View," October 2013, [Online] URL: http://goo.gl/bRcgMW.