ACCEPTED FROM OPEN CALL

Vulnerability Challenges of Software Defined Networking

Zygmunt J. Haas, Timothy L. Culver, and Kamil Sarac

As SDN becomes the technology of choice of network and service providers, it is imperative that careful attention is paid to the technology's vulnerability and proper mechanisms to protect SDN-based deployment environments. This article presents a classification of vulnerability challenges for SDN and discusses several solution strategies to address these challenges.

Abstract

As a recently introduced and rapidly accepted new network technology, software defined networking (SDN) provides network providers with operational agility and reduction in capital and operational expenses. As with all new technologies, especially one that offers rich operational flexibility, the lack of experience makes it vulnerable to potential misuse or malicious activities. As SDN becomes the technology of choice of network and service providers, it is imperative that careful attention is paid to the technology's vulnerability and proper mechanisms to protect SDN-based deployment environments. This article presents a classification of vulnerability challenges for SDN and discusses several solution strategies to address these challenges.

Introduction

From the early days of the ARPANET, the primary motivation for moving toward a packet-based network was to create a survivable system. To support this goal, the network nodes had to be autonomous. Each node of the network had its own control plane responsible for understanding and discovering its neighbors. If any changes occurred in a network (e.g., nodes/links going down), various protocols would support reconfiguration of the network. As network equipment has become more sophisticated and hence more expensive, the need for an autonomous controller for each node came into question. With 30 percent of the CPU power of a router in a data center being spent on rediscovering neighboring nodes that have not frequently changed, researchers postulated that a centralized controller may be more efficient and more cost effective than distributed controller planes across many nodes.

At this point in time, the technology of software defined networking (SDN) was born. Initiatives at universities and industry led to an effort to rethink traditional networking, with the goal of coalescing the control plane into one device. By having one control plane device handle a modern stable network, costs were reduced, because expensive advanced processing hardware was not needed on every router. The data plane devices were simplified and became cheaper, thus lowering operational expense (OpEx) and allowing the introduction of virtualized network devices, further reducing capital expense (CapEx) and OpEx.

As with most new technologies, new challenges arise. In the case of SDN, security stands at the forefront of those challenges [1]. The market forces put tremendous pressure on vendors to introduce their SDN solutions into their products and on network carriers to deploy SDN-based network services so as to differentiate themselves from competitors. This happens often without necessarily paying due attention to security implications of this new technology. In addition, introduction of virtualization technologies into networks (i.e., introduction of virtual network functions or VNFs) and their rapid adoption in SDN amplified the unforeseen threats to the networks.

The pace of change, shared resources such as storage area networks, virtual processors on the same hardware, and shared memory on the same physical hardware (the new virtualized environments) underlie these new security threats. On one hand, SDN technologies can be leveraged to support existing security services (e.g., monitoring [2], detecting [3], and mitigating [4] network security attacks via increased performance and programmability), but on the other hand, they can also introduce new security challenges that need to be carefully assessed and accounted for. However, counterintuitively, SDN also presents new opportunities [5]. Although "opportunities" may seem to be an odd term when used in the context of SDN security, let us remember that IT has already embraced virtualization for the past couple of decades, and shared lessons from the IT realm can be passed now to the network technology sharing the same environment.

One of the main challenges of security in an SDN-type environment is the concept of centralized control, in contrast to the legacy environment, where each device (e.g., router) would have its own control plane. Through centralization of network control, the environment is now more susceptible to mistakes, misuse/malicious use, and denial of service (DoS) attacks [6]. With centralized control in SDN, a successful DoS attack on the controller will likely impact a large portion of the network, as compared to a distributed controlled legacy environment, where an attack on an individual device would affect just one unit, while the remainder of the network would continue to function undisturbed. Thus, the centralized controller impacts network survivability. Consequently, the challenge is to ensure that SDN controllers are robustly designed.

The second largest challenge of SDN security is the fact that the environment is migrating from a physical environment to a virtualized one. Lack of commercial experience in using virtualization

Digital Object Identifier:
10.1109/MCOM.001.2100128

The authors are with the University of Texas at Dallas.

0163-6804/21/$25.00 © 2021 IEEE

technologies in the context of networking introduces many new security threats that must now be addressed. For instance, a physical network device being virtualized must share an environment with other virtual machines (VMs), while the other VMs in this environment could affect the performance of the now virtualized network device. In addition, all these devices provide new opportunities for bad actors to leverage these virtual network devices to attack the SDN devices. The good news is that virtualization has been around in IT environments for some time now, allowing leverage of the security knowledge in the IT domain in adapting virtualization technologies into the networking domain.

Virtualized implementations also affect the pace of change. Fifteen to 20 years ago, network changes involved hardware changes. When new network elements arrived, they were typically given a soak test. The device was turned on and monitored while network engineers made sure that no hardware or software issues existed. After this, these same devices underwent functional testing for extended periods of time. Changes to the network were slow paced, and problems were identified with the devices and their functionality. Today, with virtualization, the pace of change is drastically faster, permitting only a little time for testing. This increases the risk of introducing faulty/susceptible components into the virtualized network world. While virtualization in the IT world is mature, expanding this virtualization technology to network devices and elements almost overnight introduced rapid changes that the networking personnel were not prepared to handle.

An approach to SDN security is to consider SDN as a new environment that may be vulnerable to security threats, one that can benefit from the existing solution methodologies. Given that an SDN environment is complex, as shown in Fig. 1, potential security vulnerabilities in such an environment can be present in multiple elements, including SDN applications; SDN controllers; data plane devices including (virtual and physical) SDN switches and hybrid/legacy network devices; and the communication channels between these entities.

This article presents a classification of SDN security challenges in the various components of the SDN environment, as well as several potential solution strategies to address these challenges. The next section starts with the issues related to the use of SDN applications.

Issues Related to Use of SDN Applications

The main advantage of the SDN paradigm is that it supports rapid implementation and deployment of new network services without incurring significant changes to the existing network environment. This is achieved by implementing new services as applications on top of the controllers (Fig. 1). Such applications can interact with the network in terms of receiving necessary status information from the network and sending new forwarding rules into the network via the controller. However, this advantage also presents several security and reliability challenges for practical SDN deployments. As outlined below, these challenges stem from potential coordination and interference issues among SDN applications developed by different programmers and/or for different purposes: trust

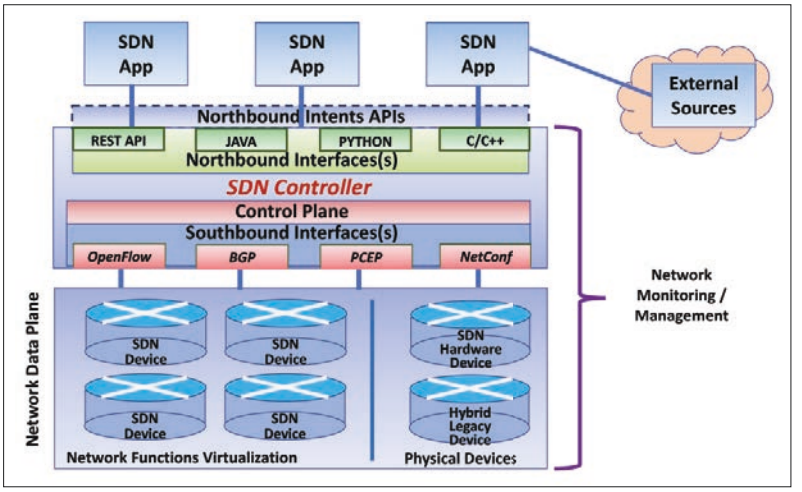

FIGURE 1. The SDN architecture.

and compliance issues related to third-party-developed SDN applications and SDN applications operated by infrastructure clients, and availability and accuracy concerns when using external data or information sources as input into the operations of various SDN applications.

Possibly Conflicting Interactions among Multiple SDN Applications: Having multiple SDN applications with different goals and objectives interacting with the underlying network through the SDN controller introduces potentially unintended consequences. Applications optimized for different goals may introduce conflicting flow rules into the SDN switches, and may result in undesired and inconsistent behavior of the network devices. Detecting the root cause of such network anomalies is often difficult and requires a comprehensive understanding of the behavior of each SDN application as well as potential interference among their actions. One possible approach to deal with this type of problem is to introduce role-based strict priority ordering for the applications. A module in the SDN controller or a separate SDN application serving as an application manager can then be utilized to monitor and order all the rules issued by other SDN applications to watch for potential conflicts and resolve them based on the priorities of the issuing applications.

Potentially Unauthorized Behavior Exhibited by SDN Applications: The SDN service model allows rapid development and rapid deployment of new network services by implementing them as SDN applications that run on top of SDN controllers. SDN applications can be developed in house by different programmers, can be offered for sale by third-party SDN application developers, or may belong to clients of a cloud service provider that offers virtual network services to their infrastructure clients. In the presence of multiple SDN applications developed or controlled by different entities, unauthorized behavior exhibited by SDN applications may become a potential malfunction challenge. The solution space to address this problem may range from program analysis (both source code and binary code analysis) to imposing certain access control restrictions (e.g., role-based access control) to building a controlled execution environment (e.g., via sandboxing or similar approaches)

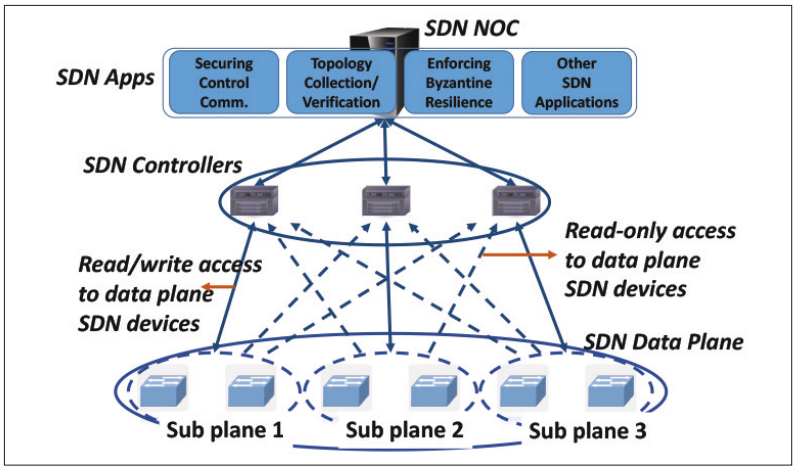

FIGURE 2. Defenses against controller compromise.

for detecting and preventing such unauthorized behavior by SDN applications [5].

Potential Security Threats Introduced by SDN Applications Interacting with Remote Third-Party Data and Information Sources: The SDN service architecture allows for introduction of value-added network services implemented by SDN applications. Some of these services may require interaction with remote data or information sources. As an example, an SDN application may subscribe to an external data service that provides information feed of natural or man-made events of significant impact on network availability or utilization (e.g., an earthquake or a developing news story). For example, SDN applications could subscribe to the United States Geological Survey to receive updates on earthquakes that could affect the availability of networks. Based on the received information, an SDN application may take the necessary actions to configure the network to deal with the now-expected imminent changes in the network (e.g., expected failure of parts of the network or expected load congestion in the network). Such third-party sources introduce another level of risk. These third-party sources could be compromised, which in turn could affect the behavior of the controller and the switches. Even though such value-added services introduced by SDN applications add significant value, the utilization of or dependence on external data/information sources may introduce additional security risks that need to be carefully addressed by operators.

Many applications and data centers have certifications, such as PCI DSS in the financial industry, HIPPA in the health care industry, SSAE 16 in data center security, service organization control (SOC) — formally SAS 70 — and certifications by International Data Center Certification. Ensuring that the source of third-party data has the right certifications will help to minimize such threats. As previously mentioned, SDN applications subscribing to information sources outside of the firewall or potentially running on remote environments pose a larger threat. To address a failed or compromised information source, if there are redundant sources of the data from third-party data sources, one could subscribe to multiple data sources to identify anomalies; that is, when the data from various sources conflicts, it provides a warning that a data source likely has been compromised. Just like circuit or power diversity, the concept here is to have data diversity to help identify threats resulting from third-party data sources being compromised.

Applications Running Outside of the SDN Controller Execution Environment: In some cases, the nature of an SDN application and its relationship to support multiple SDN controllers may necessitate the need for the controller to run on a separate server outside of the execution environment (e.g., an Open Service Gateways initiative, OSGi, container). Because the application in an OSGi container could be affected by the SDN controller application itself, the application supporting multiple controllers could be compromised. By the nature of having another server hosting an application independent of the containers hosting the SDN controllers, the risk of an attack on that single container hosting the SDN application and SDN controller can greatly increase the risk to multiple SDN controllers. For example, a service provider may have geographically distributed controllers based on latency concerns. This one application, if compromised, now threatens the entire network, not just one SDN controller covering a region.

Compromised or Unauthorized SDN Controllers

Given that the SDN controller acts as the brain of an SDN network, it becomes the primary target for an attack. If a controller serving an SDN network is compromised by an attacker, the attacker can take over the control of the entire network. In typical deployment scenarios, the controller functionality is distributed among several controllers for scalability and fault/attack resiliency. This deployment practice introduces several security requirements, including detection and prevention of unauthorized controllers and of controller hijacking, enforcing Byzantine resilience among controllers and secure access control to controllers [5]. The solution strategies (depicted in Fig. 2 and discussed below) may cover areas from network validation (via active probing-based topology mapping) to secure distributed control communication (by using cryptographically secure control communication among controllers, as well as between controllers and switches) to implementing fault tolerance for multi-controller environments (by enforcing Byzantine fault tolerance) to localizing impact of potential controller failures or compromise (by organizing a multi-level hierarchy among the controllers to achieve localization and fault isolation).

Validation of the Network Topology: The goal in this approach is to verify that the topology of an operational network in the field matches the topology design created by the network designers/operator, such that all the equipment in the field is accounted for, and that there are no unauthorized devices (controllers, switches, or other devices) planted into the network without the knowledge of the network operator. This can be achieved by developing an SDN application (Fig. 2) that will periodically solicit the topology information from the controllers and comparing the discovered topology map with the expected topology. In multi-controller environments, the monitoring application may request the topology information from each controller to cross check

the consistency of the topology map as seen by each controller.

Securing the Control Communication Path: The goal in this approach is to ensure that all control communications among controllers, between controllers and switches, as well as between SDN applications and controllers are protected against unauthorized access. This is typically achieved by using well-known cryptographically secure communication primitives among all the involved entities in the network. As an added measure, one can utilize multiple communication channels with encryption among the network entities. *OpenFlow*, in more recent versions, offers multiple communication paths of one-to-many controllers [7]. These secure communications paths ensure communication in case one path is compromised. In addition, SDN controllers can subscribe to SDN data plane devices controlled by other controllers to receive flow table changes. SDN applications could be deployed to correlate communications and ensure that the control communications path has not been compromised. If a controller is compromised or fails, these connections to other controllers can ensure that the SDN devices are managed and provide the ability to isolate the local controller failure.

Until now, the article has discussed various SDN vulnerabilities associated with attacks on the particular components of the SDN infrastructure. Another potential manner of attack on SDN involves malicious behavior that will disrupt the network operation by exploiting the way that SDN is designed to operate. Denial of service (DoS) attacks, discussed in the next section, are prime examples of such a vulnerability.

DoS Attacks on Controllers and Switch Forwarding Tables

SDN introduces potential DoS vulnerabilities targeting the controllers and/or switches in an SDN network. A potential attack on controllers involves sending new flows to an SDN managed network, thus causing a lot of control requests to be forwarded by the ingress SDN switches to the controller. The introduced control traffic attempts to saturate the bandwidth on the control channel or the processing resource at the controller site (Fig. 3). Given that switches can use per flow forwarding rules, such an attack may also aim at saturating the forwarding tables at the SDN switches. Solutions to DoS attacks can be divided into two steps: *attack detection* and *attack mitigation*. The *attack detection* approaches typically involve monitoring and modeling network traffic to identify potential attacks. These strategies include entropy-based detection, traffic pattern analysis, monitoring connection rate, using intrusion detection systems that may involve signature-based detection, and machine learning and/or big-data-analysis-based learning techniques. *Attack mitigation* approaches often closely work with attack detection schemes and involve packet dropping, port blocking, bandwidth throttling, or redirection of the offending traffic for further inspection or isolation. Additional schemes may include implementation of a moving target defense mechanism (e.g., by frequently changing the IP addresses of the potential target systems) and utilizing client challenges or cookies

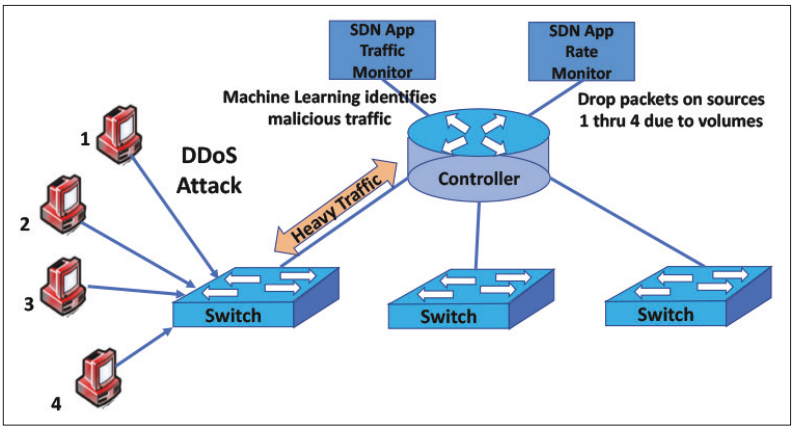

FIGURE 3. DDoS attack on SDN.

to detect IP spoofing-based attacks targeting system resources at the switches or servers as well as bandwidth resources between switches and a controller [8], as summarized below.

Traffic Monitoring/Modeling to Identify Attack Traffic: Statistical methods and machine-learning-based methods can be used to identify potential attack traffic. The recent advances in machine learning make this a new and exciting technology to assist in identifying attack traffic; this approach was not available a decade ago.

Rate Monitoring: Monitoring traffic rates for unusual spikes in traffic and implementing flow rules so as to drop packets to prevent overwhelming the resources of the switches and controllers is vital. Using statistical control methods, one could create upper and lower control limits depending on the time of day or day of the week, which are used to raise a flag when control limits have been exceeded, indicating a possible DoS or distributed DoS (DDoS) attack. Spikes or drops in traffic can be normal (e.g., terrorist event, natural disaster), and the application should take such special situations into account when analyzing the traffic volumes.

Moving Target Defense Mechanism: Dynamic flow table management and distributed control approaches can be used to reduce the impact of an attack on the control plane resources. This is an effective approach as it can keep nefarious actors confused and unable to take advantage of the vulnerability of a static environment. In particular, use of this approach makes it difficult for a malicious actor to focus on a single resource in the actor's attack.

Application Layer Defenses: Client puzzles or session cookies can be used to detect IP-spoofing-based attacks targeting system and bandwidth resources in an SDN network environment. Interactive puzzles are effective against DoS attacks when services like WiFi hotspots and corporate intranets use expensive authentication and key exchange, because interactive puzzles prevent resource depletion. Puzzles have to be solved by a human, making it extremely difficult for a scripted DoS attack to gain access. If one can randomize these puzzles through an SDN application, one can defeat a DoS attack attempting to inject packets with puzzle parameters.

Finally, DoS attacks coming from potentially malicious or compromised SDN applications should be addressed by deploying necessary pre-

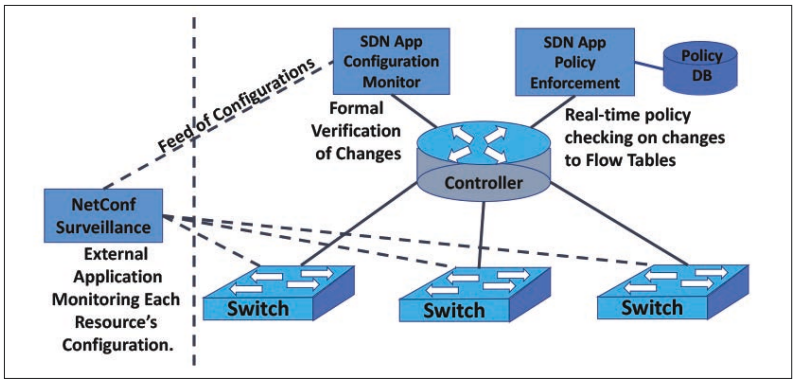

FIGURE 4. Configuration monitoring.

vention, detection, and containment approaches as discussed above.

Security Challenges Due to Configuration Issues and Policy Enforcement

The use of multiple SDN applications may introduce potential network policy conflicts. While implementing a new service, an SDN application may introduce forwarding rules that may conflict with other, already implemented forwarding rules; may be in violation of network reachability requirements; or may cause an inconsistent network state. Detecting and preventing network configuration errors has been an important domain of traditional network deployment and operation practices, and there are several approaches available to address this problem in traditional IP networks. Some potential approaches are discussed below.

Formal Verification Methods: These methods allow one to check the consistency and safety of the network configuration on the virtual resources used in SDN (Fig. 4). In addition, physical resources can be verified as well. An implementation can validate policies, ensuring that updates to configurations on the network do not violate policies. These methods can also ensure that there is consistency between multiple applications sending commands and updates to the controller. One concern that can be addressed with this approach is to help tackle issues when various applications make proposed changes to the network, where one application issues commands to the controller that conflict with another SDN application. Formal verifications will support an implementation to prove the correctness of implementations, allowing one to disable changes that present threats to the network.

Real-Time Policy Checking: Using policy-based network management promises to simplify network management and make network management much more efficient. The idea in this approach is to leverage a repository of policies used in the SDN solutions and validate each change request issued by the controller to the network devices against these policies. Such a procedure helps identify where policies are being violated by the change issued by the controller into the network devices (Fig. 4). By checking for policy compliance in real time, potential problems for the network could be identified and blocked, and alerts issued. With the growth of the variety of SDN applications running on top of the controller, real-time policy checking could be an effective safeguard against unintended consequences of such SDN applications on network operations. In addition, SDN applications with nefarious intentions could also be caught by the real-time policy checking process.

Consistent Update of Network Configuration: To ensure that a configuration update in an SDN network deployment goes as intended, one can implement simulators to determine the before and after states of the network configurations. This helps in monitoring the actual network configurations to ensure that they are consistent with the simulated configuration. If the current configuration does not match the simulated configuration, alerts can be generated that highlight the discrepancies and potential for a faulty and/or compromised environment.

Automated Back-Out: One generic approach to address potential conflicts or negative impact of any configuration changes in an operational SDN network may be to develop an automated rollback (i.e., back-out) capability in the network. In this approach, an SDN application can be created to leverage a configuration database to test and evaluate changes made by other SDN applications. If a change negatively impacts the network, the change may be rolled back automatically. The steps of this approach will include (1) creating an SDN application that will automatically evaluate and test all changes made to the network by northbound SDN applications, and (2) implementing the capability of automatically rolling back the changes that are identified as negatively impacting the network.

P4 and Its Use for Network Configuration: P4 is a language used to program an SDN hardware device handling traffic. P4 is used to program hardware and supports the OpenFlow syntax. By supporting hardware over software, there is an order of magnitude increase in performance. The latency associated with configuration and policy issues of the software-based SDN switches means that the network can be at risk for long periods of time. However, with the ability to program hardware SDN switches, high line rate switches can be configured faster, reducing the time when the network is vulnerable or at risk to an attack.

Vulnerabilities of SDN in Emerging Environments

Although the initial application domain of SDN technologies were wired data center network environments, its ability to support rapid deployment of new and innovative services has made SDN a promising technology to utilize in various other network environments. Given the rapid acceptance and adaptation of SDN technologies in various network environments, several articles have been published on security vulnerabilities related to use of SDN technologies in various application domains.

Some of the examples include use of SDN technologies to improve security and privacy protection features of 5G heterogeneous networks [9]; in building well managed, reliable, and flexible software defined vehicular networks (SDVNs) [10]; in building flexible and scalable Internet of Things (IoT) deployments by combining blockchain and SDN technologies [11]; and utilizing

SDN in battlefield network environments to deal with the heterogeneity and multi-layer distributed nature of such networks [12].

More specifically, in [8], the authors presented a classification of solution strategies against DDoS attacks on SDN networks including various machine-learning-based solutions. In [13], the authors examined the security of SDVN environments, provided a classification of security vulnerabilities in such deployments, and presented high-level solution strategies that need to be developed for securing SDVN deployments. In [14], the authors presented a survey of SDN-based solutions to address security challenges ranging from identity-based authentication to intrusion detection/mitigation to routing security to others in SDN-based IoT deployment environments. SDN technology, with its programable control of network resources, allows the physical network to be divided into a variety of logical networks through network slicing techniques [15]. Such applications introduce even further complexity in network control with corresponding security vulnerabilities.

Concluding Thoughts

SDN is a new networking technology that, as opposed to the traditional networks, provides a centralized control logic for the overall network. Among the main advantages of SDN are operational agility and reduction of operational expenses.

However, as with all new technologies, there is the danger that the limited experience with the deployment and operation of SDN will allow malicious actors to abuse its features and disrupt its operation. Although many of the threats are similar to those that the community dealt with in conventional networks, nevertheless, numerous other new potential hazards have been identified in this article and possible approaches to address those presented. The vulnerabilities of SDN are not limited to security risks, but also include failures of hardware and software components, with often severe impact on the health of the network. As the SDN technology provides a scalable mechanism to dynamically adjust network operation on a per flow basis, it supports dynamically tunable traffic analysis to detect potential vulnerabilities of end systems.

Especially exciting is the application of SDN in new and emerging environments, such as software defined vehicular networks, the Internet of Things, and new generations of cellular networks such as 5G and 6G networks. Indeed, it is envisioned that new applications of SDN will take full advantage of such opportunities. Not surprising is also the fact that some of these new technologies, such as machine learning, could be used both to create new vulnerabilities and to protect against new and existing vulnerabilities.

As SDN becomes the technology of choice of service providers, the stakes are much higher in securing the communication infrastructure, and thus it is imperative that careful attention be paid to the technology's vulnerability and that a proper mechanism be implemented to protect the infrastructure.

Acknowledgments

This work was partially funded by NSF award DGE-1433753. The work of Z.J. Haas was also partially funded by NSF award CNS-1763627.

References

[1] L. Schehlmann, S. Abt, and H. Baier, "Blessing or Curse? Revisiting Security Aspects of Software-Defined Networking," *Proc. 10th IEEE CNSM*, Rio de Janeiro, Brazil, Nov. 17–21, 2014, pp. 382–87.
[2] A. Zaalouk *et al.*, "OrchSec: An Orchestrator-Based Architecture for Enhancing Network-Security Using Network Monitoring and SDN Control Functions," *Proc. IEEE NOMS*, Krakow, Poland, 2014, pp. 1–9.
[3] C. Jeong *et al.*, "Scalable Network Intrusion Detection on Virtual SDN Environment," *Proc. IEEE CloudNet*, Luxembourg, 2014, pp. 264–65.
[4] K. Giotis *et al.*, "Combining OpenFlow and sFlow for an Effective and Scalable Anomaly Detection and Mitigation Mechanism on SDN Environments," *Computer Networks*, vol. 62, Apr. 2013, pp. 122–36.
[5] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A Survey of Security in Software Defined Networks," *IEEE Commun. Surveys & Tutorials*, vol. 18, no.1, 1st qtr. 2016, pp. 623–54.
[6] Q. Yan *et al.*, "Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges," *IEEE Commun. Surveys & Tutorials*, vol. 18, no. 1, 1st qtr. 2016, pp. 602–22.
[7] ONF, OpenFlow Switch Specification, v, 1.4.1 (Protocol Version 0x05), ONF TS-024, Mar. 26, 2015.
[8] K. Kalkan, G. Gur, and F. Alagoz, "Defense Mechanisms against DDoS Attacks in SDN Environment," *IEEE Commun. Mag.*, vol. 55, no. 9, Sept. 2017, pp. 175–79.
[9] X. Duan and X. Wang, "Authentication Handover and Privacy Protection in 5G HetNets Using Software-Defined Networking," *IEEE Commun. Mag.*, vol. 53, no.4, Apr. 2015, pp. 28–35.
[10] Z. He, J. Cao, and X. Liu, "SDVN: Enabling Rapid Network Innovation for Heterogeneous Vehicular Communication," *IEEE Network*, vol. 30, no. 4, July/Aug. 2016, pp. 10–15.
[11] P. K. Sharma *et al.*, "DistBlockNet: A Distributed Blockchain-Based Secure SDN Architecture for IoT Networks," *IEEE Commun. Mag.*, vol. 55. no. 9, Sept. 2017, pp. 78–85.
[12] G. M. Leal *et al.*, "Empowering Command and Control through a Combination of Information-Centric Networking and Software Defined Networking," *IEEE Commun. Mag.*, vol. 57, no. 8, Aug. 2019, pp. 48–55.
[13] A. Akhunzada and M. K. Khan, "Toward Secure Software Defined Vehicular Networks: Taxonomy, Requirements, and Open Issues," *IEEE Commun. Mag.*, vol. 55, no. 7, July 2017, pp. 110–18.
[14] K. Kalkan and S. Zeadally, "Securing Internet of Things with Software Defined Networking," *IEEE Commun. Mag.*, vol. 56, no. 9, Sept. 2018, pp. 186–92.
[15] A.A. Barakabitze *et al.*, "5G Network Slicing Using SDN and NFV: A Survey of Taxonomy, Architectures and Future Challenges," *Computer Networks*, vol. 167, Feb. 11, 2020, pp. 1–40.

Biographies

Zygmunt J. Haas [S'84, M'88, SM'90, F'07] received his Ph.D. from Stanford University, California, in 1988. From 1988 until 1995, he worked for AT&T Bell Laboratories, after which he joined Cornell University, Ithaca, New York. Since 2013, he has been a professor and Distinguished Chair at the University of Texas at Dallas, Richardson. He is a recipient of numerous awards and distinctions, including IET Fellow, EAI Fellow, and IEEE ComSoc Recognition Awards. His research interests include wireless networks, network security, and modeling of complex systems.

Timothy L. Culver [S'84, M'99] received his undergraduate degree from Baylor University, Waco, Texas, and his M.S. in Engineering and M.B.A. from Southern Methodist University, Dallas, Texas. He did postgraduate Ph.D. research at both Baylor University iand Walden University, Minneapolis, Minnesota. He has five patents and is a lecturer in the Computer Science Department at the University of Texas at Dallas. He has been a Co-Chair and panelist lead for IEEE NFV/SDN conferences.

Kamil Sarac [S'99, M'02, SM'12] received his B.S. degree in computer engineering from Middle East Technical University, Ankara, Turkey, in 1994 and his Ph.D. degree in computer science from the University of California Santa Barbara in 2002. Currently, he is a professor of computer secience at the University of Texas at Dallas. His research interests include computer networks, network security, and Internet measurements.

As SDN becomes the technology of choice of service providers, the stakes are much higher in securing the communication infrastructure, and thus it is imperative that careful attention be paid to the technology's vulnerability and that a proper mechanism be implemented to protect the infrastructure.