Benefits and the Security Risk of Software-defined Networking

feature

Traditionally, organizations increase their network bandwidth by focusing on buying more hardware. This approach does not always work, and it could be a costly mistake if the additional network resources are not fully utilized. As technology evolves, history finds a way to repeat itself. From the days of mainframe-based network protocols, such as Systems Network Architecture, the transition was made to the adoption of the universally accepted Transmission Control Protocol/Internet Protocol (IP) network protocol—a transition caused by the introduction of the personal computer (PC), which uses client-server computing technology. Now, with the disruptive and fast-paced changes from PCs to mobile devices, such as smart phones, which are enabled by virtualization and cloud-computing models, it looks as though the future of networking is increasingly going to rely on automated software. One might wonder about network evolution and how a modern network infrastructure will respond to the ever-changing demands of end users, commercial businesses and government regulators.

The growth in connected devices that could reside anywhere in the world has increased the complexity and difficulty of managing them and the related network traffic. There are very high costs associated with manually reconfiguring these devices for any required changes. Moreover, it is often difficult and sometimes almost impossible to reconfigure the traditional network in a timely manner to react to human errors and/or malicious events. Software-defined networking (SDN) makes use of virtualization to greatly expand network efficiency and, thus, simplifies the management of those consolidated resources and provides solutions for increased capacity without breaking the bank.[1]

Benefits of SDN

What is SDN? Unlike traditional network design, SDN design is a paradigm shift that uses software-based controls to simplify the execution of policies with a centralized controller. It separates the data and control functions of networking devices, such as routers and switches, with a well-defined application programming interface.[2]

SDN architecture is logically separated into three planes: the application plane, the control plane and the data plane. The application plane incorporates SDN applications, which communicate the network requirements to the SDN controller. In turn, the software-based SDN controller interprets these requirements and executes the actual network policy from the control plane, which determines how data should flow from network devices. The SDN controller is the core of the SDN architecture, handling all complex functions and translating requirements into specific low-level rules. Finally, the data plane contains network devices, e.g., routers and switches, which execute the data flow once given permissions from the SDN controller. In essence, SDN decouples the network control and forwarding functions, enabling the network protocol to become directly programmable and the underlying infrastructure to be abstracted for applications and network services.[3]

Since SDN uses a centralized controller with software applications, one of the biggest benefits of implementing SDN is its flexibility. Because the SDN controller assumes most complex functions (e.g., managing network intelligence and monitoring the network behavior in real time), the network

Tony Wang
Is the principal of Williams Adley's IT risk management practice. He has more than 15 years of experience in evaluating internal controls, information systems and software development. Prior to joining Williams Adley, Wang held various management positions at BDO USA, Ernst & Young and Lockheed Martin, where he served clients in various industries, with a strong focus on government, health care, manufacturing, technology, nonprofit and financial services. His areas of specialty include engineering process auditing, integrated financial auditing, information assurance, IT security auditing, software development life cycle assessment, and systems integration and assessment. In addition to client responsibilities, Wang has spent a significant amount of time working on business development, campus recruiting, counseling, and developing and conducting training for IT auditors.

> A centralized controller provides flexibility, programmability and a high return on investment (ROI) with its simplified network design.

devices just need to accept orders from the SDN controller. This eliminates the need for the network devices to understand how to execute data flow based on different communication protocols from various vendors. As a result, this gives network administrators great flexibility to configure, manage, secure and optimize network devices.[4]

> **“As companies adopt cloud technology, SDN will be able to simplify the overall network design by leveraging virtualization to automate network management operations.”**

Furthermore, since the SDN controller is the brain of the SDN architecture, it is much easier to modify software programs/applications than manually reconfigure every single network device. This benefit alleviates the need for enterprises to purchase additional expensive network devices to meet ever-changing business needs. Therefore, SDN becomes a very cost-effective solution. In fact, SDN is designed to eliminate the dependence on the vendor locked-in network approach (i.e., vendors have their own proprietary management console and set of commands), which frees the enterprise to drive innovation and enhance network interoperability. As a result, it will greatly increase the ROI.

A majority of traditional network is managed using management consoles with a command-line interface that requires a lot of manual effort from network administrators.[5] As companies adopt cloud technology, SDN will be able to simplify the overall network design by leveraging virtualization to automate network management operations.

The Security Risk of SDN

Many security issues related to the traditional network architecture also apply to the SDN architecture. Unfortunately, the new features that provide great flexibility, real-time programmability and simplified controls through the centralized SDN controller also introduce new security challenges. In fact, SDN is exposed to various sources of security risk from its network architecture design perspective, which includes the control plane, application plane and data plane layers.

One of the most significant security risk factors is the possibility of a compromised SDN controller attack at the control plane layer. Due to the centralization design of the SDN, the SDN controller becomes the brain of the SDN architecture. Attackers can focus on compromising the SDN controller in an attempt to manipulate the entire network.[6] If the attacker successfully gains access, the compromised SDN controller can be used to direct the network devices it controls (e.g., switches) to drop all incoming traffic or launch serious attacks against other targets, such as sending useless traffic to a victim to deplete its resources.[7] To mitigate this security risk, it is critical to harden the operating system that hosts the SDN controller and prevent unauthorized access to the SDN controller. Furthermore, the control plane layer is susceptible to a distributed denial-of-service (DDoS) attack. SDN switches may cause the SDN controller to be flooded with many queries that may potentially cause a delay or drop of queries. One possible defense against a DDoS attack is to implement multiple physical SDN controllers

> If attackers compromise the SDN controller, they can hack the SDN applications to manipulate security applications to reprogram the network traffic flow through the SDN controller.

instead of just one. When switches are connected to multiple SDN controllers, one of these controllers can act as the master of the switches. When this master controller needs to process a high load of queries, it can direct the load to other lightly loaded controllers to be the master for some of its assigned switches. This keeps the load balanced among the SDN controllers, which mitigates DDoS attacks.

At the data plane layer, switches are vulnerable to denial-of-service (DoS) attacks as well. A malicious user can flood the switches with large payloads, causing legitimate packets to be dropped when a switch's buffering capability is exceeded. There are many ways to address this attack, including proactive rule caching, rule aggregation and decreasing the switch-to-SDN-controller communication delay. Also, increasing the switch's buffering capability can mitigate the risk of a DoS attack.[8]

Communicating messages between the control plane layer and the data plane layer is subject to man-in-the-middle attacks. The attacker can potentially modify rules sent from the SDN controller to switches to take control of the switches. One of the most effective solutions to such attacks is to encrypt the messages with the use of digital signatures for securing and proofing the integrity and authenticity of the messages.

The real-time programmability is also open to serious vulnerability at the application plane layer. Specifically, if the attacker can hack the SDN security applications, it can manipulate the network traffic flow through the SDN controller. If the SDN applications are compromised, the whole network is, too.[9] To effectively mitigate such security risk, it is critical that security coding practices be enforced with comprehensive change management and integrity check processes as part of the software development life cycle.

Conclusion

Server virtualization, mobility and cloud computing are becoming the new norm to meet changing business needs. As these technologies evolve, the traditional network architecture is starting to fall short of meeting the significant network demands. The SDN architecture provides a virtualized network that transforms today's network into flexible and programmable platforms. The future of networking will rely more and more on software, and SDN, in turn, will become the new norm for networks. On the other hand, there is critical security risk that needs to be addressed regarding the SDN controller and applications before the SDN can be securely deployed.

Endnotes

[1] Underdahl, B.; G. Kinghorn; *Software Defined Networking for Dummies*, John Wiley & Sons, USA, 2015

[2] Stallings, W.; "Software-Defined Networks and OpenFlow," *The Internet Protocol Journal*, vol. 16, no. 1, March 2013, p. 2-14

[3] Open Networking Foundation, *https://www.opennetworking.org/sdn-resources/sdn-definition*

[4] Open Networking Foundation, *Software-Defined Networking: The New Norm for Networks*, 13 April 2012, *https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf*

[5] Kim, H.; N. Feamster; "Improving Network Management With Software Defined Networking," *IEEE Communications Magazine*, vol. 51, iss. 2, February 2013, p. 114-119

[6] Open Networking Foundation, *Principles and Practices for Securing Software-Defined Networks*, January 2015, *https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/Principles_and_Practices_for_Securing_Software-Defined_Networks_applied_to_OFv1.3.4_V1.0.pdf*

[7] Mehiar, D.; B. Hamdaoui; M. Guizani; A. Rayes; "Software-defined Networking Security: Pros and Cons," *IEEE Communications Magazine*, vol. 53, iss. 6, June 2015

[8] Dabbagh, M.; B. Hamdaoui; M. Guizani; A. Rayes; "Software-Defined Networking Security: Pros and Cons," *IEEE Communications Magazine*, vol. 53, iss. 6, May 2015

[9] Lim, A.; "Security Risks in SDN and Other New Software Issues," RSA Conference 2015, July 2015