ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 12, December 2017

Software-Defined Networking

[1]Dhruv Kumar

[1]Department of Computer Science and Engineering, Galgotias University, Yamuna Expressway Greater Noida,
Uttar Pradesh

[1]dhruv.kumar@Galgotiasuniversity.edu.in

Abstract: Software-Defined Networking (SDN) is a network infrastructure technique that allows the network to be managed, or' programmed' intelligently and centrally utilizing software applications. It allows providers to control the entire network efficiently and holistically, irrespective of the network technology that underlies it. The Software Defined Networking (SDN) drive is a gravitational one. In the networking world, there are few who have avoided their effect. As the advantages of network availability and programmability of network devices are being debated, the query might be raised as to who exactly would benefit? Will it be the user of the network or will it really be the attacker of the network? When SDN products and applications hit the market, SDN protection needs to be brought up on the agenda. This paper presents a comprehensive survey of the research that has been carried out to date on security in software-defined networking. This addresses both the protection benefits to be gained from the use of the SDN platform and the security issues raised by the system. A collection of findings and recommendations for potential avenues in study are provided by categorizing the existing work.

Keywords: SDN, Security, Challenges faced in SDN security.

INTRODUCTION

With a number of SDN-enabled devices in development and production, software-defined networking (SDN)[1] is quickly progressing from dream to practice. The convergence of independent control and data plane connectivity and network programmability, debated for a long time in the research world, has found a commercial application in cloud computing and virtualization technologies. SDN's benefits across different scenarios (e.g. the business, the datacenter etc.) and across different infrastructure networks have already been demonstrated, e.g. Google. There are however obstacles for SDN deployment of a full-scale carrier network. One key area, which merely begins to receive the attention it requires, is that of SDN health.

The SDN design is used with the introduction of a highly reactive security control, review and response system to enhance network protection. The key to that system is the central controller. Traffic analysis or anomaly-detection methods deployed in the network generate data related to security, which is transmitted to the central controller on a regular basis. Frameworks is operated at the controller to evaluate this input from the entire network and to compare it.
Based on the analysis, in the form of flow rules, new or updated security policies is propagated over the network. This consolidated approach may effectively speed up the control and containment of the threats to network security. Yet the same unified management and programmability characteristics identified with the SDN architecture pose network security issues. A prime example is an increased potential for Denial-of-

Service (DoS)[2] attacks due to the centralized controller and limitation of the flow-table in network devices. Another issue of concern is confidence, centered on the network's flexible programmability; both between software and controllers, and controllers and network devices.

In the literature a variety of approaches have been suggested for these SDN security challenges. Which vary from controller duplication schemes to processes of verification, to protocol conflict resolution. Likewise, a range of suggestions have been made to take advantage of the SDN architecture to enhance network security.

In this document an overview of SDN's security challenges is discussed. The individual security issues are categorized to be affected or targeted according to the SDN layer. Then it discusses and categorizes the proposed and emerging solutions to these challenges. The requirement for further work in order to establish a secure and robust SDN is clearly identified from the gap between the problems and existing research. Without a considerable increase in security focus, SDN will not be able to support the evolving capability associated with, for example, Network Functions Virtualization (NFV).

SECURITY ANALYSES OF SDN:

The essential properties of a secure communications network are: secrecy, transparency, knowledge security, authentication and non-repudiation. Security professionals will encrypt the records, network infrastructure (e.g. devices), and communications transfers across the network to provide a network safe from malicious attack or accidental harm. To ensure that network security is sustained, the changes to the network architecture introduced by SDN need to be assessed.

In an early version of what is known today the protection implications of a different control and forwarding system were explicitly addressed. Their SANE model, introduced in 2006, was based on a theoretically unified controller responsible for host authentication and implementation of policies. This was considered an extreme approach at the time of its proposal, which would require a radical change to the networking infrastructure and end-hosts, which might be too restrictive for some companies.

Ethane continued SANE's research but adopted a methodology that needed less modification to the initial network. This managed the network by using two components; a centralized controller responsible for global policy compliance, and ethane switches, which essentially redirected packets in a flow table based on the rules. A streamlined management of the network has allowed the data and control plane to be segregated to create more programmability. Though the Ethane design offered us a closer look at what would become of SDN and Open Flow, it suffered from a range of drawbacks. One of these is that application traffic has the potential to compromise network policy. Applications are used in today's SDN architecture to provide various services, such as Network Functions Virtualization (NFV)[3] for example. Application compromises could potentially breach the entire network.

In terms of the specific security problems in SDN from the SDN architecture viewpoint (Fig. 1), we should define challenges associated with each layer of the system: device, control and data planes, and on the interfaces between these levels. Recently, a number of security analyzes were conducted which found that the altered elements or relationships between elements in the SDN framework introduced new vulnerabilities that were not present before SDN.

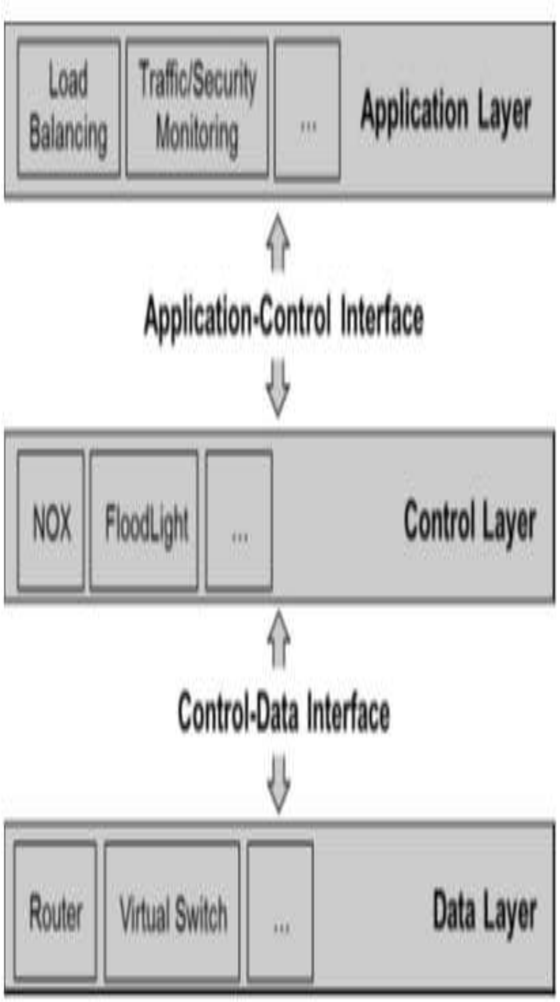


Figure 1: SDN Functional Architecture

The specification of the Open Flow switch describes the use of transport layer safety (TLS) with mutual authentication between controllers and their switches. The security feature is optional, however, and not specifying the TLS standard. A risk evaluation of Open Flow reflects on the absence of TLS acceptance by major vendors, and the probability of DoS assaults. The authors found that lack of TLS usage could result in fraudulent insertion of rules and modification of rules.

A high-level analysis of SDN security as a whole. We believe that new threats needing new solutions are being added due to the nature of the centralized controller and the network's programmability. We suggest a number of techniques to tackle the various threats like duplication, diversity and modules that are stable. Finally, it also analyzed the research network and testbed, ProtoGENI. The authors found that when using the ProtoGENI network, numerous attacks between testbed users along with destructive spreading and flooding attacks to the wider internet were feasible.

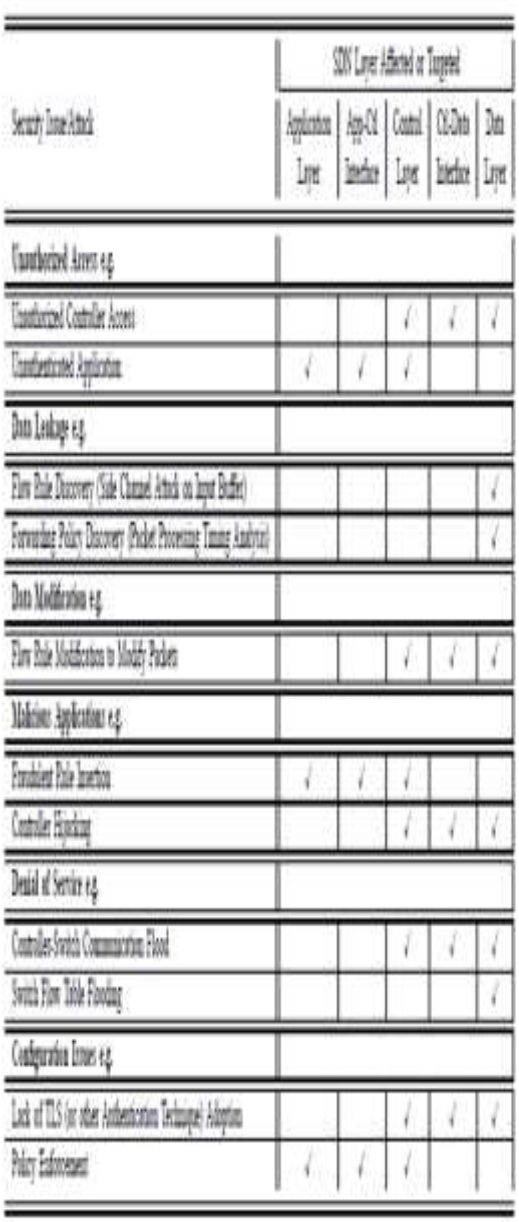
The results of these analyzes show the range of security issues linked to the SDN framework. A categorization of security issues with the SDN is presented in Table I. A connection is drawn between the problem / attack type (e.g. unauthorized access) and the issue / attack-affected SDN layer / interface. Table I identifies the control and data layers as clear targets of attack. This reflects the major distinctions between the traditional network and the SDN; that of the centralized control element and the altered data path elements for programmability support.

While this report points to security issues relevant to the control and data layers, there was limited field work to tackle the challenges. Indeed, as detailed in the next section, greater attention has been given to exploring the potential improvements to be derived from the SDN framework in network safety.

SECURITY ENHANCEMENT USING SDN:

The software-defined network architecture introduces potential for innovation in network usage. Combining global or network-wide view with network programmability, for example, supports a process of harvesting intelligence from existing intrusion detection systems (IDS) and intrusion prevention systems (IPS)[4], followed by network analysis and centralized reprogramming. This strategy will make the SDN more resilient than conventional networks for malicious attack.

Table 1: Categorization of the security issues



Security Issue/Attack	SDN Layer Affected or Targeted				
	Application Layer	App-Ctl Interface	Control Layer	Ctl-Data Interface	Data Layer
Unauthorized Access e.g.					
Unauthorized Controller Access			√	√	√
Unauthenticated Application	√	√	√		
Data Leakage e.g.					
Flow Rule Discovery (Side Channel Attack on Input Buffer)					√
Forwarding Policy Discovery (Packet Processing Timing Analysis)					√
Data Modification e.g.					
Flow Rule Modification to Modify Packets			√	√	√
Malicious Applications e.g.					
Fraudulent Rule Insertion	√	√	√		
Controller Hijacking			√	√	√
Denial of Service e.g.					
Controller-Switch Communication Flood			√	√	√
Switch Flow Table Flooding					√
Configuration Issues e.g.					
Lack of TLS (or other Authentication Technique) Adoption			√	√	√
Policy Enforcement	√	√	√		

SECURITY CHALLENGES WITH SDN:

While safety was recognized as an advantage of the SDN framework, fewer solutions are available to tackle the challenges of securing the SDN network.

SDNs give us the ability to easily program the network, and to create dynamic flow policies. Yes, it is this benefit that contribute to protection flaws as well. It is important that network security strategy is applied within this dynamic environment. Model-checking becomes an important step in detecting policy inconsistencies from multiple applications or multi-device installations. Model checking combined with symbolic execution may be used to test the correctness of Open Flow applications.

Binary Decision Diagrams can also be used within a single flow to check for intra-switch misconfigurations. Flow Checker abuses Flow Visor, allowing separation by splitting the network resources into slices. Son et al. suggests Flover, which uses assertion sets and modulo theories to verify flow policies, while VeriFlow studies real-time invariant verification. An additional layer that sits between the SDN controller and the devices in the network intercepts flow rules before they reach the network. While VeriFlow boasts low checking process latency, it cannot handle multiple controllers.

Proposals for support in developing stable SDNs are however restricted. One noteworthy addition is Fresco; which offers an Open Flow Defense Application Development Platform integrating Fort NOx; a kernel [5]for defense regulation. The concept behind FRESCO is to allow the fast design and development of specific security modules that may be implemented as an Open Flow framework. This program implements the compliance engine Fort NOx which manages possible conflicts with the implementation of laws. If a rule conflict arises as a consequence of a new Open Flow rule allowing or disabling a prohibited / allowed existing rule, then the new rule will be accepted or rejected to the established infringing law supplier, based on the author's level of security authorization. Although Fort NOx provides numerous components needed to enforce security, the authors feel there is still a lot of work to be done to offer a comprehensive suite of applications.

LITERATURE REVIEW

This software implements FortNox compliance engine which handles possible conflicts with law enforcement. If a code conflict arises as a consequence of a new OpenFlow rule allowing or disabling a prohibited / allowed existing rule, then the new rule will be accepted or rejected on the basis of the author's level of security authorisation to the specified infringing law supplier. Although FortNox provides numerous components necessary to enforce security, the authors feel that there is still a great deal of work to be done to offer a comprehensive suite of applications[6]. This work describes the operation of the proposed architecture and summarizes the opportunity in a more efficient and flexible way with SDN to achieve network security[7]. This paper explores the initiation of DDoS attacks on SDN, and strategies for DDoS attacks in SDN. The inconsistent interaction regarding SDN and DDoS attacks was not well discussed in previous works, to the best of our knowledge[8]. This paper focuses on the analysis and categorization of a number of relevant research works aimed at achieving SDN promises. We first provide a summary on SDN origins and then explain the core design of SDN and its key components. Afterwards, we discuss current SDN-related taxonomies and suggest a taxonomy that classifies the research works examined and reflects on relevant research directions[9].

CONCLUSION

In software-defined networking there are two think tanks on security. The first is that significant improvements in network security can be achieved by simultaneously exploiting the SDN introduced programmability and centralized view of the network. The second is that the network is vulnerable to a range of new threats by these same two SDN characteristics. We also described the SDN security challenges in this report, and provided a comprehensive review of the SDN security research work to date. Our analysis identifies that there is yet more to be done regardless of your school of thought; more untapped potential, and more unresolved challenges.

REFERENCES

[1] M. Olsson, S. Sultana, S. Rommer, L. Frid, and C. Mulligan, “Architecture Overview,” in *EPC and 4G Packet Networks*, 2013.

[2] N. Bahl, A. K. Sharma, and H. K. Verma, “On Denial of Service Attacks for Wireless Sensor Networks,” *Int. J. Comput. Appl.*, vol. 43, no. 6, pp. 43–47, 2012, doi: 10.5120/6111-8348.

[3] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, “Network function virtualization: State-of-the-art and research challenges,” *IEEE Commun. Surv. Tutorials*, 2016, doi: 10.1109/COMST.2015.2477041.

[4] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, “A survey of intrusion detection techniques in Cloud,” *Journal of Network and Computer Applications*. 2013, doi: 10.1016/j.jnca.2012.05.003.

[5] Q. Yan, F. R. Yu, Q. Gong, and J. Li, “Software-defined networking (SDN) and distributed denial of service (DDOS) attacks in cloud computing environments: A survey, some research issues, and challenges,” *IEEE Communications Surveys and Tutorials*. 2016, doi: 10.1109/COMST.2015.2487361.

[6] B. A. A. Nunes, M. Mendonca, X. N. Nguyen, K. Obraczka, and T. Turletti, “A survey of software-defined networking: Past, present, and future of programmable networks,” *IEEE Commun. Surv. Tutorials*, 2014, doi: 10.1109/SURV.2014.012214.00180.

[7] O. Flauzac, C. Gonzalez, and F. Nolot, “New security architecture for IoT network,” 2015, doi: 10.1016/j.procs.2015.05.099.

[8] A. Blenk, A. Basta, M. Reisslein, and W. Kellerer, “Survey on network virtualization hypervisors for software defined networking,” *IEEE Communications Surveys*

and Tutorials. 2016, doi: 10.1109/COMST.2015.2489183.

[9] Y. Jarraya, T. Madi, and M. Debbabi, “A survey and a layered taxonomy of software-defined networking,” *IEEE Communications Surveys and Tutorials*. 2014, doi: 10.1109/COMST.2014.2320094.