Software-Defined Network Security over OpenStack Clouds: A Systematic Analysis

Nicolas P. Lane[1][a], Guilherme P. Koslovski[1][b], Maurício A. Pillon[1][c], Charles C. Miers[1][d] and Nelson M. Gonzalez[2][e]

[1]*Graduate Program in Applied Computing (PPGCA), Santa Catarina State University (UDESC), Brazil*
[2]*IBM Watson Research Center, Yorktown Highs, NY, U.S.A.*
nicolas@colmeia.udesc.br, {guilherme.koslovski, mauricio.pillon, charles.miers}@udesc.br, nelson@ibm.com

Keywords: Cloud Computing, OpenStack, Security, Openflow.

Abstract: Cloud computing infrastructure is an enticing target for malicious activity due to its network and compute capacity. Several studies focus on different aspects of cloud security from the client (tenant) side, leaving a gap regarding the cloud provider's infrastructure perspective. To address this gap, this study conducts a systematic review of the literature on OpenStack, the most adopted open source cloud operating system. We present a qualitative assessment of security vulnerabilities related do Openflow usage on OpenStack network management. Based on this analysis we identify a critical vulnerability which affects the cloud infrastructure via Software-Defined Networks. This reveals the urge for having more studies focusing on the provider's infrastructure side and associated tools and technologies.

1 INTRODUCTION

Cloud computing is a well-established concept of resource and service provisioning to which new features and technologies are constantly being incorporated. Cloud computing depends on several enabler technologies, particularly ones related to virtualization. In that sense, several solutions have been developed over the years to manage the cloud infrastructure and provide services to customers.

A cloud operating system encompasses features both to create this cloud infrastructure, including the setup of virtualized resources, and interfaces to allow user access. The ecosystem of cloud operating systems comprise proprietary and open source solutions. In terms of security, still one of the main concerns of cloud computing, proprietary solutions can only be inspected from its interfaces since there is no access to the source code, therefore hindering the process of investigating vulnerabilities (CSA, 2017). In contrast, open source solutions allow a thorough investigation of internal mechanisms.

OpenStack, one of the most prominent open source cloud platforms, stands out in terms of source code auditing, project maturity, versatility, and constant updates (Wen et al., 2012; Mullerikkal and Sastri, 2015; Vogel et al., 2016; Iqbal and Dagiuklas, 2017). In its 18th release (codename Rocky) (OpenStackRocky, 2018a), OpenStack organises its networks using three NDs (Network Domains) (Rosado and Bernardino, 2014): management, guest, and public. These NDs comprise networks managed by four SecNDs (Security Network Domains): management, guest, public, and data (OpenStack, 2018a). Networking is a fundamental cloud building block which depends on a combination of technologies such as VLAN, L2 tunnelling, and SDN (Software-Defined Network).

In this paper we present the results of a systematic review of OpenStack security literature. We focus on OpenStack's SDN-related mechanisms due to vulnerabilities identified in the past (Thimmaraju et al., 2016) and its relevance to the current OpenStack architecture. We identified a general lack of studies on cloud networking performance and security, in particular regarding cloud operating systems. Also, most existing studies focus on a client perspective. Only one of the surveyed studies analyses performance from an infrastructure provider perspective (Sciammarella et al., 2016), and none of them

[a] https://orcid.org/0000-0001-8050-1638
[b] https://orcid.org/0000-0003-4936-1619
[c] https://orcid.org/0000-0001-7634-6823
[d] https://orcid.org/0000-0002-1976-0478
[e] https://orcid.org/0000-0003-3780-9025

Table 1: Studies on OpenStack SecNDs (Security Network Domains).

Study	Addressed network security domain				Focused upon	
	Public	Guest	Management	Data	Client	Provider
(Lar et al., 2011)	✓	✗	✗	✗	✓	✗
(Ristov et al., 2014)	✓	✓	✓	✗	✓	✗
(Carlsson, 2015)	✓	✗	✗	✗	✓	✗
(Felsch et al., 2015)	✓	✗	✗	✗	✓	✗
(Astrova et al., 2016)	✓	✗	✗	✗	✓	✗
(Thimmaraju et al., 2016)	✗	✗	✓	✓	✗	✓
(Qiu et al., 2017)	✓	✗	✗	✗	✓	✗
(Benjamin et al., 2017)	✓	✗	✗	✗	✓	✗
(Thimmaraju et al., 2018)	✗	✗	✓	✓	✗	✓

address provider security. This is especially important for IaaS (Infrastructure-as-a-Service) since it is the provider's responsibilities to implement protection mechanisms and deploy secure infrastructure services (CSA, 2017).

The paper is organised as follows: Section 2 presents the motivation for the investigation; Section 3 presents OpenStack networking and SDN; Section 4 presents a vulnerability vector; Section 5 analyses a particular set of vulnerabilities identified in this study; and Section 6 presents related work.

2 LITERATURE REVIEW

Our initial investigation efforts revealed a limited number of studies related to OpenStack SecNDs, as presented in Table 1. In each cell of the table the symbols ✓ and ✗ indicate if a particular aspect is addressed or not by each studied work. Most of the studies addressed the public SecND, which is focused on the client perspective of the cloud to show potential customers the benefits of cloud computing. Other client-focused studies have also been published in less quantity over the years addressing other SecNDs (Venzano and Michiardi, 2013; Sciammarella et al., 2016). However, only two works covered provider's cloud infrastructure (Thimmaraju et al., 2016; Thimmaraju et al., 2018). This lack of provider-side studies is not desirable, especially considering that vulnerabilities (and potential mitigation solutions) typically are a product of a wide spectrum investigation of the mechanisms involved. One example is the SDN-related vulnerability (Thimmaraju et al., 2016) which could be exploited to compromise the OpenStack's cloud infrastructure. Moreover, the consequences of a provider-side vulnerability may spread through all infrastructure, therefore compromising all its users. This motivated a more thorough research of infrastructural security vulnerabilities focusing on the provider's mechanisms.

3 OpenStack AND SDN

OpenStack's main networking service is called Neutron. Further technologies can be added using its API (Application Programming Interface). Examples include FWaaS (Firewall-as-a-Service), SDN, LBaaS (Load-Balancer-as-a-Service), and VPNaaS (Virtual-Private-Network-as-a-Service) (OpenStack-Rocky, 2018b). Particularly, SDN applied to OpenStack enhances the cloud network infrastructure management by providing elasticity, flexibility, and network programmability. SDN separates switching logic in the data plane and control plane, centring its management over a single dedicated controller or a cluster of distributed controllers that act as a single entity through the use of location transparency (Kuźniar et al., 2015; Yu et al., 2015; Masoudi and Ghaffari, 2016; Singh and Jha, 2017).

The SDN abstraction comprises three planes. The application plane allows applications to request network services to the control plane. The controller communicates with the data plane and the switches will forward the data at the physical layer based on flow tables, which are managed by the SDN controller. The SDN abstraction is solely based on virtualization technologies. These three planes communicate using two specific types of APIs: the northbound API, for bidirectional communication between the control plane and application plane; and the southbound API, for bidirectional communication between control plane and data plane.

There are several SDN standards, such as OpenFlow and ForCES. OpenStack uses OpenFlow, the most prominent open source implementation for the southbound API. The specification depends on the support for the SDN controller solution and for the vSwitch solution.

OpenStack's network infrastructure is organised in different NDs as presented in Figure 1. The public ND comprises the external network, the public network, and the API network. The management ND comprises the storage network and the management network. The guest ND only contains the guest network.

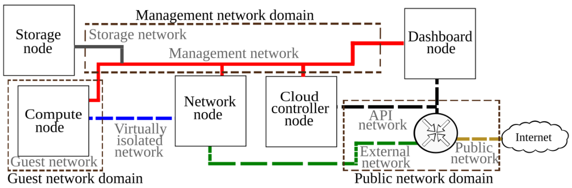


Figure 1: Network infrastructure of an OpenStack cloud.

NDs are used for servicing tasks and also for internal communication rules between cloud services required for operability and portability. In Figure 2 the OpenStack NDs are presented with some examples of main services and their relation to OpenStack SecNDs.

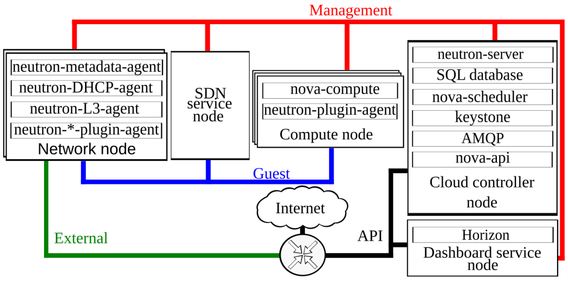


Figure 2: OpenStack services associated to NDs.

Each ND has specific security mechanisms defined by the SecND, which also specifies if communication between two NDs is possible and how it should be conducted while addressing trust levels required by each ND. The services associated to the SecNDs are presented in Figure 3.

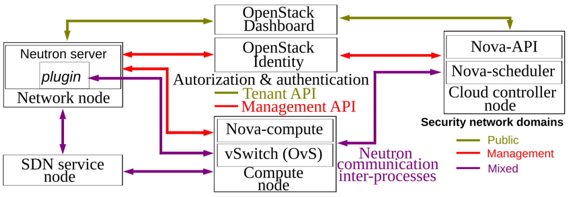


Figure 3: OpenStack services associated to security NDs.

4 OpenStack SDN Vulnerability Vector

Based on the initial literature review we conducted systematic research. We verified that OpenStack uses Open vSwitch among several SDN and vSwitch controller technologies. A SDN controller must be able to communicate with Neutron through REST (Representational State Transfer) in order to be supported by OpenStack. Currently there are three main solutions: Floodlight, OpenDaylight, and Ryu. Each solution has specific attributes, such as support for different OpenFlow specifications and for different implementations of TLS (Transport Layer Security). There are also similar aspects, such as open source projects which support the same communication protocol.

Regarding vulnerabilities, we adopted two selection criteria to identify and evaluate SDN vulnerabilities in OpenStack: 1)The results reported by the authors of each study, focusing on main findings and the potential impact of these vulnerabilities; and 2)Our own analysis of the implications of the vulnerabilities over the affected technologies regarding implementation, development, and official documentation from each of the described projects.

Since OpenFlow is a common denominator for these solutions, our systematic review used the following search string: *"OpenFlow vulnerability OR OpenFlow vulnerabilities OR OpenFlow flaw OR OpenFlow flaws OR OpenFlow vulnerability* OR OpenFlow vulnerabilities* OR OpenFlow flaw* OR OpenFlow flaws*"*. We adopted several scientific search tools, such as Web of Knowledge, Engineering Village, and Scopus SciVerse. Table 2 presents details of the results regarding the search tools used. A hyphen indicates that the access to the tool was restricted. Symbols ✓ and ✗ indicate service availability. This study was conducted until December 23, 2019.

Table 2: Scientific research tools and results.

Scientific research tool	**Results**	**Promising results**	**Service tool status**
Web of Knowledge	-	-	✓
Engineering Village	-	-	✓
Scopus SciVerse	-	-	✗
IEEE Xplore	29	11	✓
ACL DL	5.552	0	✓
Science Direct	127	1	✓
Springer Link	27	2	✓
BASE	2	0	✓
Scirus	-	-	✗
InSpire HEP	0	0	✓
CiteSeerX	1.787.998	0	✓
DBLP	0	0	✓
Ingenta Connect	0	0	✓
Google Scholar	100	17	✓

Due to the lack of studies addressing OpenStack provider perspective, we also mapped all technologies with known vulnerabilities used in OpenStack. The results are presented in the mind map in Figure 4, which shows five major classes of potentially vulnerable technologies. Our study focuses on SDNs since there is a significant number of security reports related to it. Also, the figure addresses related technolo-

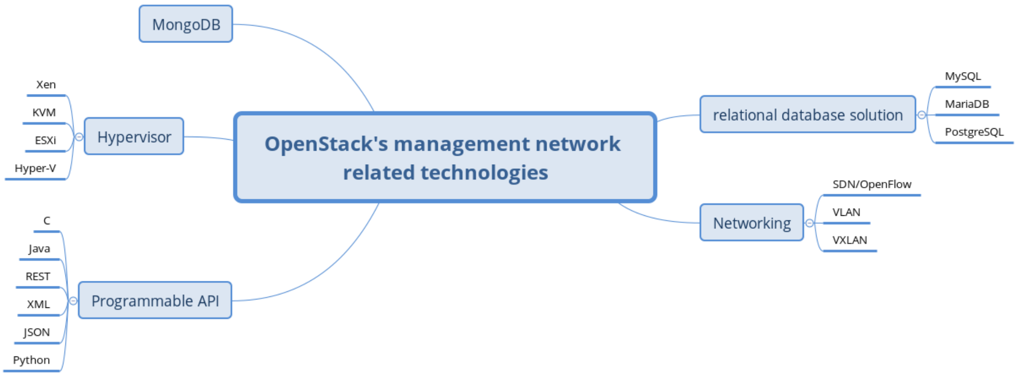
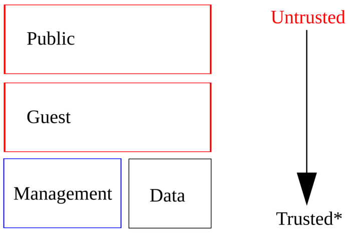


Figure 4: Technologies associated to OpenStack.

gies specifically from the management network since it is a very restricted network in terms of access from the provider's network. To the best of our knowledge, these aspects have been not properly addressed by previous works. One possible reason for this fact is that this level of SecND is typically regarded as secure, and therefore does not require further investigation, or because it does not directly face users, so the main threat that could compromise the internal infrastructure are malicious insiders. However, the level of trust delegated to this SecND heavily depends on existing bridges it has with other SecNDs, which lowers its trust level. This is illustrated by Figure 5.



*Demands verification - some
data require additional security

Figure 5: Trust between different OpenStack SecNDs, Adapted from (OpenStack, 2018b).

5 ANALYSIS

Following the methodology and preliminary results presented in the previous sections, we identified several security vulnerabilities associated to technologies used by OpenStack. Also, we identified chain reactions that might occur if a vulnerable technology is used by another more complex technology that is used by OpenStack, which consequently can be used to exploit OpenStack infrastructure. This led to the definition of a vulnerability vector which allows to evaluate all the identified vulnerabilities. This vector affects OpenStack infrastructure solely through its SDN-related mechanisms. The results are presented in Table 3. Table 3 shows a total of 11 vulnerabilities. We provide a qualitative classification based on the severity and potential impact on OpenStack infrastructure:

- *High Severity*: Vulnerability with potential to compromise the SDN controller, subduing Neutron's architecture and allowing an attacker to take control of the OpenStack infrastructure or highly compromise its operability (Thimmaraju et al., 2016). From the control plane the attacker acquires privileged and unrestricted access over OpenStack infrastructure.
- *Average Severity*: Vulnerability with potential to obtain sensitive information and even modify it, which combined with other attacks has the potential to compromise specific areas of the SDN architecture. This access, however, is limited in terms of privilege to a specific node.
- *Low Severity*: Vulnerability with potential to acquire information about the SDN architecture. This typically is an initial step before launching a more dangerous attack, so although it might have low severity, it must be stopped before the attacker acquires enough information about the system to then compromise it with more severity.

Nine of the identified vulnerabilities are related to ambiguities in OpenFlow specification (from version 1.0.0 on 2009 to 1.5.1 on 2015), most due to the lack of a clear feature definition. The other two vulnerabilities identified in our study are due to the implementation of unspecified services in regard to the OpenFlow specification. One vulnerability only affects OpenDaylight, while the other affects all controllers tested. The most severe vulnerability is related to the usage of the TLS 1.0 implementation for SDN controller communication.

Table 3: Vulnerability vector over OpenStack's SDN architecture.

				Open source affected solutions			
				SDN controllers			vSwitches
				Floodlight 1.2	OpenDaylight Fluorine	Ryu 2.28	OvS 2.10.90
Index	Implicates	Vulnerability alias	Severity				
1	**TLS**	*Man-in-the-Middle*	*High*	✓	✓	✓	✓
2		*Switch authentication*	*High*	✓	✓	✓	✗
3	**Flow table**	*Flow Table overlap*	*Average*	✓	✓	✓	✗
4	**Excessive requests towards the SDN controller**	*Widespread DoS– I*	*Low*	✓	✓	✓	✗
5		*Low network intrusiveness DoS – II*	*Average*		✓		✓
6		*Undefined authorization criteria for distributed SDN architecture*	*High*	✓	✓	✓	✗
7		*Undefined granular access/management for distributed SDN architecture*	*High*	✓	✓	✓	✗
8		*Switch tampering*	*High*	✓	✓	✓	✓
9	**Information disclosure**	*Aggregation link information disclosure – I*	*Low*	✓	✓	✓	✗
10		*RTT information disclosure – II*	*Low*	✓	✓	✓	✗
11	**Host tracking service**	*Host hijacking attack*	*Average*	✗	✓	✗	✗

Table 3 also reveals that the majority of the identified vulnerabilities is related to excessive number of requests to the SDN controller. An alternative OpenFlow specification was produced in 2013 (ONF, 2013) to address this particular issue. However, the security aspects are still not being adequately addressed in a timely fashion once discovered. A significant number of security studies (NWG, 2012; Chung et al., 2013; OTHMAN, 2013; Dover, 2013; Benton et al., 2013; Kloti, 2013; Kandoi and Antikainen, 2015; Tiwari et al., 2014; Hong et al., 2015; Brooks and Yang, 2015; Kandoi and Antikainen, 2015; Chellani, 2016; Thimmaraju et al., 2016; Cui et al., 2016; Agborubere and Sanchez, 2017; Dargahi et al., 2017; Mostovich et al., 2017; Li et al., 2017; Zhou et al., 2018; Mutaher et al., 2018; Bhatia et al., 2018; Ilyas and Khondoker, 2018) identified vulnerabilities in OpenFlow and we observed that they usually exploit the same vulnerability or a small variation over the years until present time (2019). The vulnerability vector we engineered only maps vulnerabilities relevant to the OpenStack SDN context, which operates along with the OpenFlow southbound protocol of communication. Several of these vulnerabilities were first discovered circa 2013 and they still affect the OpenFlow protocol.

Figure 6 summarises our findings in terms of number of vulnerabilities associated to each solution or technology. Most vulnerabilities are associated to OpenFlow's specification. If the specification addressed vulnerabilities and issues already identified by previous works, changed the TLS cryptography standard to the required, and if OpenStack SDN related technologies implemented the appropriate TLS standard at least four out of eleven (36%) vulnerabilities would be mitigated, several with high severity.

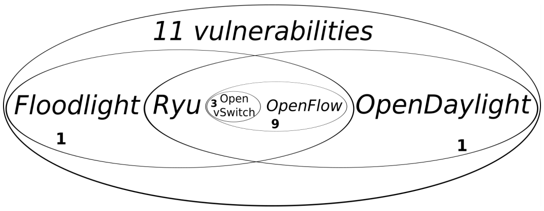


Figure 6: Vulnerability distribution across technologies.

6 RELATED WORK

Due to the lack of studies focusing on the security of management networks, we also analysed works related to networking performance. (Venzano and Michiardi, 2013) evaluate OpenStack's capacity to execute applications with high data volume. The authors did not address aspects related to security. (Sciammarella et al., 2016) provide a characterisation of management network traffic. The authors also did not address security of these networks.

(Thimmaraju et al., 2016) provide a proof of concept for vulnerabilities in Open vSwitch applied to OpenStack. The authors exploited this vulnerability over the SDN data plane, compromising the management network domain and therefore acquiring unrestricted access to the cloud infrastructure. Finally, (Thimmaraju et al., 2018) propose a new threat model for virtual switches to satisfy the requirements identified in their previous study.

7 CONSIDERATIONS & FUTURE WORK

This work proposes a vulnerability vector addressing security aspects related to technology and implementation with focus on the SDN technologies used by OpenStack. We identified eleven vulnerabilities with different levels of severity, which are related to the potential damage they might inflict to OpenStack's infrastructure. Those with the highest severity can be exploited to fully compromise the infrastructure. Moreover, several of these vulnerabilities can be mitigated by adopting newer and more secure methods to protect the traffic between different planes and security domains. We intend to extend our research to other open source cloud platforms.

ACKNOWLEDGEMENTS

The authors would like to thank the support of the LabP2D/UDESC, providing its facilities and resources to the accomplishment of this research. FAPESC supported LabP2D infrastructure.

REFERENCES

Agborubere, B. and Sanchez, E. (2017). OpenFlow Communications and TLS Security in Software-Defined Networks. pages 560–566.

Astrova, I., Koschel, A., and Henke, M. L. (2016). IaaS Platforms: How Secure are They? In *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pages 843–848.

Benjamin, B., Coffman, J., Esiely-Barrera, H., Farr, K., Fichter, D., Genin, D., Glendenning, L., Hamilton, P., Harshavardhana, S., Hom, R., Poulos, B., and Reller, N. (2017). Data Protection in OpenStack. In *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*, pages 560–567.

Benton, K., Camp, L. J., and Small, C. (2013). OpenFlow vulnerability assessment. page 151. ACM Press.

Bhatia, S., Nathani, K., and Sharma, V. (2018). Review on Software-Defined Networking: Architectures and Threats. In Bhateja, V., Nguyen, B. L., Nguyen, N. G., Satapathy, S. C., and Le, D.-N., editors, *Information Systems Design and Intelligent Applications*, volume 672, pages 1003–1011. Springer Singapore, Singapore.

Brooks, M. and Yang, B. (2015). A Man-in-the-Middle Attack Against OpenDayLight SDN Controller. In *Proceedings of the 4th Annual ACM Conference on Research in Information Technology*, RIIT '15, pages 45–49, New York, NY, USA. ACM.

Carlsson, A. (2015). Model of network attack on the cloud platform OpenStack. In *2015 Second International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S T)*, pages 245–247.

Chellani (2016). Enhancing Security in OpenFlow.

Chung, C., Khatkar, P., Xing, T., Lee, J., and Huang, D. (2013). NICE: Network Intrusion Detection and Countermeasure Selection in Virtual Network Systems. *IEEE Transactions on Dependable and Secure Computing*, 10(4):198–211.

CSA (2017). Security Guidance For Critical Areas of Focus in Cloud Computing v4.0.

Cui, H., Karame, G. O., Klaedtke, F., and Bifulco, R. (2016). On the Fingerprinting of Software-Defined Networks. *IEEE Transactions on Information Forensics and Security*, 11(10):2160–2173.

Dargahi, T., Caponi, A., Ambrosin, M., Bianchi, G., and Conti, M. (2017). A Survey on the Security of Stateful SDN Data Planes. *IEEE Communications Surveys Tutorials*, 19(3):1701–1725.

Dover (2013). A denial of service attack against the Open Floodlight SDN controller.

Felsch, D., Heiderich, M., Schulz, F., and Schwenk, J. (2015). How Private is Your Private Cloud?: Security Analysis of Cloud Control Interfaces. pages 5–16. ACM Press.

Hong, S., Xu, L., Wang, H., and Gu, G. (2015). Poisoning Network Visibility in Software-Defined Networks: New Attacks and Countermeasures.

Ilyas, Q. and Khondoker, R. (2018). Security Analysis of FloodLight, ZeroSDN, Beacon and POX SDN Controllers. In Khondoker, R., editor, *SDN and NFV Security*, volume 30, pages 85–98. Springer International Publishing, Cham.

Iqbal, M. and Dagiuklas, A. (2017). Infrastructure as a Service (IaaS): A Comparative Performance Analysis of Open-Source Cloud Platforms.

Kandoi, R. and Antikainen, M. (2015). Denial-of-service attacks in OpenFlow SDN networks. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 1322–1326.

Kloti, R. (2013). OpenFlow: A security analysis. pages 1–6. IEEE.

Kuźniar, M., Perešíni, P., and Kostić, D. (2015). What You Need to Know About SDN Flow Tables. In Mirkovic,

J. and Liu, Y., editors, *Passive and Active Measurement*, Lecture Notes in Computer Science, pages 347–359. Springer International Publishing.

Lar, S. u., Liao, X., and Abbas, S. A. (2011). Cloud computing privacy & security global issues, challenges, & mechanisms. In *2011 6th International ICST Conference on Communications and Networking in China (CHINACOM)*, pages 1240–1245.

Li, C., Qin, Z., Novak, E., and Li, Q. (2017). Securing SDN Infrastructure of IoT–Fog Networks From MitM Attacks. *IEEE Internet of Things Journal*, 4(5):1156–1164.

Masoudi, R. and Ghaffari, A. (2016). Software defined networks: A survey. *Journal of Network and Computer Applications*, 67:1–25.

Mostovich, D., Fabrikantov, P., Vladyko, A., and Buinevich, M. (2017). High-level vulnerabilities of software-defined networking in the context of telecommunication network evolution. In *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pages 184–186, St. Petersburg and Moscow, Russia. IEEE.

Mullerikkal, J. P. and Sastri, Y. (2015). A Comparative Study of OpenStack and CloudStack. In *2015 Fifth International Conference on Advances in Computing and Communications (ICACC)*, pages 81–84.

Mutaher, H., Pradeep, K., and Abdul, W. (2018). OpenFlow controller-based SDN: Security issues and countermeasures. *International Journal of Advanced Research in Computer Science*.

NWG (2012). Security Analysis of the Open Networking Foundation (ONF) OpenFlow Switch Specification.

ONF (2013). OpenFlow specification 1.3.3.

OpenStack (2018a). OpenStack Docs: Networking services.

OpenStack (2018b). OpenStack Docs: Security boundaries and threats.

OpenStackRocky (2018a). OpenStack Docs: Rocky.

OpenStackRocky (2018b). OpenStack Docs: Service Extensions.

OTHMAN (2013). Securing Distributed Control of Software Defined Networks. *IJCSNS International Journal of Computer Science and Network Security , VOL. 13 No .9*.

Qiu, Y., Shen, Q., Luo, Y., Li, C., and Wu, Z. (2017). A Secure Virtual Machine Deployment Strategy to Reduce Co-residency in Cloud. In *2017 IEEE Trustcom/BigDataSE/ICESS*, pages 347–354.

Ristov, S., Gusev, M., and Donevski, A. (2014). Security Vulnerability Assessment of OpenStack Cloud. In *2014 Sixth International Conference on Computational Intelligence, Communication Systems and Networks*, pages 95–100.

Rosado, T. and Bernardino, J. (2014). An Overview of Openstack Architecture. In *Proceedings of the 18th International Database Engineering & Applications Symposium*, pages 366–367, New York, NY, USA. ACM.

Sciammarella, T., Couto, R., Rubinstein, M., Campista, M., and Costa, L. (2016). Analysis of Control Traffic in a Geo-Distributed Collaborative Cloud. pages 224–229.

Singh, S. and Jha, R. K. (2017). A Survey on Software Defined Networking: Architecture for Next Generation Network. *Journal of Network and Systems Management*, 25(2):321–374.

Thimmaraju, K., Shastry, B., Fiebig, T., Hetzelt, F., Seifert, J.-P., Feldmann, A., and Schmid, S. (2016). Reigns to the Cloud: Compromising Cloud Systems via the Data Plane. page 15.

Thimmaraju, K., Shastry, B., Fiebig, T., Hetzelt, F., Seifert, J.-P., Feldmann, A., and Schmid, S. (2018). Taking Control of SDN-based Cloud Systems via the Data Plane. pages 1–15. ACM Press.

Tiwari, V., Parekh, R., and Patel, V. (2014). A Survey on Vulnerabilities of Openflow network and its impact on SDN/Openflow controller. *World Academics Journal of Engineering Sciences*, 01(01):1005.

Venzano, D. and Michiardi, P. (2013). A Measurement Study of Data-Intensive Network Traffic Patterns in a Private Cloud. In *Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*, pages 476–481, Washington, DC, USA. IEEE Computer Society.

Vogel, A., Griebler, D., Maron, C. A. F., Schepke, C., and Fernandes, L. G. (2016). Private IaaS Clouds: A Comparative Analysis of OpenNebula, CloudStack and OpenStack. In *2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, pages 672–679.

Wen, X., Gu, G., Li, Q., Gao, Y., and Zhang, X. (2012). Comparison of open-source cloud management platforms: OpenStack and OpenNebula. In *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*, pages 2457–2461.

Yu, C., Lumezanu, C., Sharma, A., Xu, Q., Jiang, G., and Madhyastha, H. V. (2015). Software-Defined Latency Monitoring in Data Center Networks. In Mirkovic, J. and Liu, Y., editors, *Passive and Active Measurement*, Lecture Notes in Computer Science, pages 360–372. Springer International Publishing.

Zhou, Y., Chen, K., Zhang, J., Leng, J., and Tang, Y. (2018). Exploiting the Vulnerability of Flow Table Overflow in Software-Defined Network: Attack Model, Evaluation, and Defense. *Security and Communication Networks*, 2018:1–15.