Software Defined Networks' Security: An Analysis of Issues and Solutions

Egbenimi Beredugo Eskca, Omar Abuzaghleh, Priya Joshi, Sandeep Bondugula, Takamasa Nakayama, Amreen Sultana

Abstract-Lately, Software Defined Networks (SDN) has received a lot of attention as a new technology which provides more flexibility than conventional network. It is the decoupling of the data plane from the control plane. In the implementation of SDN, three outstanding benefits readily come to mind; network flexibility, speedy service provisioning and efficiency as well as lower operating expense. SDN also has vast potential of programmability, configurability and manageability from its unique character of centralized software control. The exponential growth of mobile devices and content, server virtualization and the introduction of cloud services are among the key computing trends which need new networking architecture. Furthermore, handling today's "big data" requires extensive parallel processing on thousands of servers, all of which needs secure connections to each other. These massive, complex, and sensitive data and user requirements beckons on a new improved, dynamic and dependable network infrastructure and architecture which is promised on the centralized control based architecture of SDN. This paper attempts to delineate the strengths and weaknesses of SDN. We commence with a listing of identifiable security threats and breaches of SDN. The paper then makes an analysis of previously outlined solutions to identifiable security issues of SDN. We venture further into the horizon of the unknown to predict and identify new security breaches and threats, as well as areas of inherent weakness in the overall SDN architecture and infrastructure. Possible solutions to the identified issues are proffered and analyzed by the paper. In view of the limitations of this research, the paper prescribes possible positions for future researchers to adopt, in order to shed more light to the pertinent security issues of SDN.

Index Terms- CRUTIAL, DDOS, FortNox, OpenFlow, OpenSec, OpenVSwitch, SDN, Software Defined Networking, STRIDE

1 Introduction

The apparent rigidity and lack of flexibility and programmability of legacy network architecture has been the concern of many networking enthusiast over the years. The necessity to overcome these lapses in today's network has been the focus of many industry and academic research efforts. Chief amongst these are work carried out on programmable networks such as, active networks, programmable ATM networks and on proposals for control and data plane decoupling such as the network control points (NCP) and routing control platform (RCP) [1]. Consequent upon these remarkable contributions is the requirement to deliberately isolate the functionality of the data plane from that of the control plane. The current standard, in which the data forwarding functions and the control functions are built into a single hardware, is the reason for and the basis of the lack of flexibility and programmability of the current network structure. The argument has been made and reasonably so, that if the data and control functions are decoupled and isolated from the single plane on which they hitherto reside, networks would be flexible and programmable in a manner that would overcome most of the setbacks of today's network architecture: complexity, management nightmare, heterogeneity and manual configuration.

Software defined networks (SDN) is the proposed solution to the current issue of the de facto network architecture. With SDN the data plane is separated from the control plane, and network control can be centrally administered.

Our research effort anchors on the need for a thorough analysis of foreseeable security challenges, and their proposed solutions, as well as identifies new security challenges in SDN and proffer possible solutions to these challenges. We commence with a brief literature review and trends on SDN, and conclude with possible solutions, and suggestions on the way forward regarding the research for a secure software defined network.

2 Related Works

The key ingredients of a secured communication network are: confidentiality, integrity, data availability, ease of authentication and non-repudiation [1], [2], [3]. Sandra et al sited the contribution of Ethane et al and how the Ethane architecture of SDN extended the proposal of an existing architecture-SANE. They pointed out that, although the Ethane architecture outlined a more detailed analysis on what SDN and openFlow would later become,

the architecture suffers from a number of major flaws; including but not limited to the fact that application traffic could comprise network policy.

The introduction of a software extension called FortNox, to the openFlow controller provided an initial standard to measure SDN networks on the basis of their security performance [4]. FortNox provided a role based security system for openFlow with three levels of access. It is a scheme in which the security precedence level of the role inserting application decides, which role takes precedence over another.

In analyzing SDN security, focus is primarily anchored on the Distributed denial of service and how it can be used to target SDN based networks. In some papers [5], [9], the purpose and the strategies employed by attackers to exploit the vulnerability of SDN based networks are clearly pointed out. The purpose of these attacks is to deplete bandwidth and exhaust network resource. A major strategy of DDOS attacks outlined by the authors is the use of program snippets called botnets, which are injected into a machine in the target network, from whence the attack is initiated.

The OpenSec's innovative approach allows operators to customize the security of the network using human-readable policies and how the controller reacts automatically when malicious traffic is detected [10]. An Orchestrator-based architecture that utilizes Network Monitoring and SDN Control functions to develop security applications was also proposed for OpenSec' security implementation of SDN[11].

The concept of proactive-reactive recovery presents a design paradigm for a generic proactive-reactive recovery service that can be integrated in any intrusion-tolerant system [12]. Infrastructure protection, which uses the proactive-reactive recovery service, is the main beneficiary of this scheme.

Minzhe et al [13] focused on the controller placement for network resilience improvement in SDN. The authors addressed three salient points; analyzing the impact of controller placement on SDN resilience from the perspective of interdependent networks, defining a new resilience metric based on the cascading failure analysis on the interdependence graph, and proposing a partition and selection approach to controller placement for improving the resilience of SDN networks.

FortNox is represented again as a new security policy enforcement kernel. FortNOX in this context, is viewed as an extension to the open source NOX OpenFlow controller[14].

New architectural proposition of SDN composed of three main components: OpenFlow switches as forwarding elements, Domain Controllers (DC) that serve local requests and applications and Parent controllers (PC) to perform control functions for local requests, was analyzed [15].

3 ANALYSES

Kloti[4] commenced with and extensive exposition of the STRIDE methodology. The methodology clearly outlined the components of network security threats and attacks, against which these threats are protected. According to the STRIDE analysis, the primary components of SDN threats include, spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege. These threats concerns network security properties such as authentication, integrity, non-repudiation, confidentiality, availability and authorization.

Kloti alluded to data flow diagrams which are graphical representation of data flow in a program. Data flow diagrams present a pictorial representation of data models such as; data flows, data stores, processes, interactors and trust boundaries. Kloti's analysis termed network connections as an element of data flow, while data stores represent database tables. Trust boundaries refer to the components of SDN that separate different levels of trust, while interactors depict the various data producers and consumers of a network system such as users. The various vulnerability components and target elements of Kloti's analysis are represented in table 1.0.

As seen in the table 1.0 above, processes are most affected of all component types. Processes are vulnerable to all attack types according to the STRIDE analysis, while interactors are the least vulnerable.

4 SDN SECURITY THREAT MODELING USING ATTACK TREES

Attack trees and graphs have been used by various authors [4], [5], [16] to model network security threats. Although most of the work was related to traditional networks, the models are applicable to the fundamental structure and architecture of SDN [5]. Kloti[4] and Sandra et al[2] provided graphical analysis as well as mathematical models and algorithms of attack tree modeling of network security threats. An algorithm for building attack tree as referenced by Kloti [4]is provided below.

-Define the attack objective, which becomes the root node.
-Recursively divid this objective into prerequisite objectives.
-decompose the attack structure in detail e.g. analyzable quantitative forms (degree of difficulty in deciphering 2048 RS key)
- Assign values on getting to the leaf nodes e.g. cost or

Table 1.0 Components vulnerability and attack types

Attack type	Security property	Data flow	Data stores	Processes	Interactors
Spoofing	Authentication	No	No	Yes	Yes
Tampering	Integrity	Yes	Yes	Yes	No
Repudiation	Non-repudiation	No	No	Yes	Yes
Information disclosure	Confidentiality	Yes	Yes	Yes	No
Denial of service	Availability	Yes	Yes	Yes	No
Elevation of privilege	Authorization	No	No	Yes	No

difficulty of execution.
-Propagate values up the tree and make calculations based on specific model.

Sequel to the analysis of the above algorithm, Kloti pointed out a danger in the quantitative modeling of the attack tree, reasoning that the uncertainty of the leaf node values would rather support a variable probabilistic model rather than a precise quantitative model. The attack tree algorithm should be considered a systematic descriptive model rather than a concrete quantitative model he inferred.

The attack types are analyzed as follows:

a. Spoofing: With spoofing, an attacker pretends to be a legitimate user of a network resource. This could be achieved by forging fake MAC, or IP address. Attackers can fake ARP packets as well in their attempt to fool the system into believing that they are legitimate users with legitimate network resource request.
b. Tampering: In tampering attacks, an attacker attempt to get the system to modify a given data item from its original form to a form that meets the attacker's need. This could be achieved by getting the controller to install flow rules intended to modify or falsify data packets or flow counters[17].
c. Repudiation: When a generated content is not traceable to an originator, repudiation attack is possible. With a repudiation attack, an attacker falsifies packet source address, and sends packets to a desired destination. In doing so, the receiving system cannot accurately determine the source of the received packets.

d. Information disclosure: Being in possession of information you are not permitted to have is generally referred to as information disclosure. In the context of SDN, this could imply side channels attacks intended to reveal extended information about the openflow system.
e. Denial of service: DOS attacks are designed to limit the system's ability to transmit and received data in a normal and predictable manner. This is achieved via the use of techniques designed to deplete bandwidth and system resources. This is where openflow is most vulnerable presenting its largest surface to attack ratio. The requirement of SDN that packets must be sent to the controller on a regular basis, presents potential opportunities for denial of service attacks [4], [5], [8], [9], [16].
f. Elevation of privilege: This consists of the ability of an attacker ascribing to himself the opportunity to perform system operations that he otherwise would be unable to perform. The only feasible way to achieve this kind of attack in SDN is to assume control over the controller. This is considered a potentially difficult task due to the use of SSL.

Because of its relevance to the security of SDN, this paper presents and examine the approach of Ashraf et al [5] in combating DOS and DDOS attacks. The authors outlined two types of intrusion detection techniques.

a. Signature detection technique: This involves the use of special algorithm to search network traffic for the presence of packets sequences that are known to be malicious.
b. Anomaly detection: In this technique, the baseline of the normal network behavior is predefined. Events in the detection engine are triggered when a network behavior outside the acceptable threshold is detected. Techniques for Anomaly detection
    a. Statistical analysis
    b. Machine learning

Although the two types of anomaly detection techniques are equally important, particular attention is given to the analysis of the machine learning approach. The machine learning approach is typically made of the following techniques.

a. Neural networks: This is based on the techniques used by biological nervous system to process information. It consists of a collection of processing elements aimed at transforming a set of input to a set of output. Neural networks mainly work on the lines of classification. Then classify attack pattern, attack type as well as the normal network behavior. Neural networks could be efficient in what they do after a period of training.
b. Support vector machines: This is among the most common machine methods for classifying machine learning tasks. The technique involves the use of a set of marked categories of training examples. Technique specific algorithms are used to construct a model that could determine if a new example falls within any of the previously marked categories of examples. The classification algorithm involves:
    - Determine an input space X for each network connection.
    - Select n attribute characteristics.

- Use the vector x (one-dimensional) to describe a network connection as follows:
- x = {xl, x2, ... , xn }, where xi , i = 1,2, ... n, denote the i characteristic value of the sample x.
- define Y = (+ 1,-1) (to represent normal or abnormal connection)
- If Y = + 1 then connection is normal
- if Y= -1 then connection is abnormal

Because of its promising results in the learning of small samples, a support vector machine is a good choice in intrusion detection in SDN.

c. Genetic algorithms: Genetic algorithm according to [5] likens network attributes such as; service, flags, login status and super user attempts to individual chromosomes in genes. This technique involves search methods that provide approximate solution to an optimization problem. A profile is created for a normal and acceptable behavior. Base on the created profile, the genetic algorithm makes the decision of which network behavior is normal or dodgy. The technique works more efficiently with known attack pattern, but is less popular with new and evolving attack patterns.

d. Fuzzy logic: Because of the variability of the anomalous conditions of possible network intrusions, the notion of fuzzy logic is well suited to the design of intrusion detection logic in network security [5], [18]. Fuzzy logic allows an object to fit into different classes at the same time. Although security applications that implements fuzzy logic have registered satisfactory degree of success, it's tendency to consume extensive network resources and the extended time needed for training, are major disadvantages in the application of fuzzy logic designs in SDN security

e. Bayesian networks: The Bayesian network scheme is built on the naïve Bayesian algorithm which is used primarily for learning tasks, where training set with target class is provided. The aim is to classify an unknown pattern whose class is unknown. Below is a sample demonstration of the Bayesian theorem.

Given the values of attribute (a, a2, ... , an) which describe the sample.

Cmap = argmax C} E CP C} aI, a2 ... ... a)the expression can be rewritten using Bayesian theorem as

Cmap = argmax C} E C(aI, a2 ... an I C})P(C}) . . . .. (1)

Each of the p(C) is estimated simply by counting the frequency of occurrence Cj of the target class in the training sample.

f. Decision trees (DT): Decision trees uses algorithm based deductive inference and predictive modeling techniques to estimate target functions that produces discrete values. Intrusion detection system in SDN is a classification problem since connections or users, needs to be identified either as a valid or normal connection, user or as one of the classified attack types. DT is widely used in the areas of machine learning, data mining and statistics to solve classification based problems. DT constructs easily interpretable models that assist network security operatives to inspect and edit network records and reports [4].

The separation in SDN of the functional network units as discussed in the introduction of this paper is key to the desired flexibility of SDN, breaking the network control problem into tractable pieces, and making it easier to create and introduce new abstractions in networking; thus simplifying network management and facilitating network security management[1]. SDN provides an application programming interface (API) allowing a network's data plane to be altered by external applications. This concept is two-sided with respect to security because it enables both new security mechanisms and new threats. First SDN provides a vulnerable network security functions by design [3]. It is thus the opinion of this paper that a deliberate focus on security is essential if SDN is to take its place as the network architecture of the future. Consequently, various security working groups have been set up for this purpose. Notable among these is the one in the Open Network Foundation (ONF)[19].

Following and exhaustive analysis, Phillip Porras et al [14] proposed an idea using FortNOX as an extension of NOX OpenFlow controller. The main role of FortNOX is to provide non-bypassable policy-based flow rule enforcement over flow rule insertion requests from OpenFlow application. Minzhe Guo and Probir Bharracharya [13] in a rather sensational analysis, argues that controller placement is one of the critical problems in SDN design. They focused on network resilience improvement in SDN for their controller placement research. One of the salient characteristics of SDN is centralizing the control logic they inferred.

The research of Neda Beheshti and Ying Zhang [20] pointed out the vulnerability of SDN, and their idea of using failover is very attractive to the practical deployment of SDN. Their failover scheme certainly strengthens the failure of communication between switches and controller.

In SDN centralized control model, logically centralized SDN controllers are potentially subject to a different set of risks and threats compared to conventional network architectures. Since the controller is centralized, it will be a potential single point of attack and failure. This paper considered the Automated malware quarantine (AMQ) proposed by Cohn et al[21] as a viable solution to protecting network devices. Once AMQ detects insecure network devices, it isolates it before it adversely affects the network.

Lara et al. analyzed and proposed OpenSec [10], which is based on OpenFlow security framework which allows network security operators to create and implement policies in human-readable language. By making use of this method a user can describe the flow in terms of OpenFlow matching fields and can decide on which security service can be applied to which flow e.g.(deep packet inspection, intrusion detection and spam detection etc). Decision is also made on which security level, e.g. alert, blocking or quarantining should be applied if any malicious content is detected. In order to evaluate the flexibility, accuracy and scalability of the framework the authors have implemented OpenSec in GENI

test bed which uses virtual nodes and OpenVSwitches to perform deep packet inspection, intrusion detection and network quarantining to secure the web server from the network scanners.

Paulo Sousa et al. proposed a complementary approach which enhances proactive-reactive recovery mechanisms [12]. They designed a device called CIS which is an abbreviation for CRitical UTility InfrastructurAL resilience (CRUTIAL) Information Switch, an intrusion tolerant firewall for critical infrastructures. Their design was based on hybrid distributed system model. The experimental results provided proved that the service is effective in the presence of powerful DoS attacks which may be triggered by external hosts or internal compromised replicas.

Zhiyuan Hu et al.[22] addressed solutions for the open flow security and also proposed a comprehensive security architecture which enables security services like enforcing mandatory network policy correctly, to receive the network policy from the north bound API securely and to enhance the packet data scan detection to mitigate some attacks like worms. This architecture will also guide the naïve developers to implement security functions in developing the SDN controller.

Christopher C. Lamb et al. [23] defined the trust relationship between the various entities which are based on attributes like confidentiality, integrity, availability, non-repudiation and authentication. They described the implications of security characteristics on these entities. They also analyzed the variations between SDNs and other domains with active trust research, describing why those differences were important as well as their implications.

Adopting a somewhat divergent approach to their analysis some researchers [7], [11] , [24], [26] reasoned that the most efficient algorithms that could proactively combat the prevalent resource attacks such as DOS and DDOS are those that are built into the core functionality of the OpenFlow system. In addition to the forgoing, the requirement to enhance SDN security by building additional security apparatus on top of the controller was also emphasized.

To overcome the security issues which are not covered by existing internet systems such as centralized control,

Orchestrator-Based Architecture was proposed by some authors[11].The architecture utilizes network monitoring and SDN control functions to develop security applications.

Braga et al [24]Presents a lightweight method for DDOS attack detection based on traffic pattern analysis. This method includes Self-Organizing Map (SOM) algorithm to classify network traffic as normal and abnormal. SOM is an unsupervised artificial neural network trained with features of the traffic flow. SOM works as follows:

1. Initialization: at the beginning of the process all neuron vectors have their synaptic weights randomly generated.

2. Sampling: a single sample x is chosen from the entry pattern space, and fed to the neuron grid.

3. Competition: based on the minimum Euclidean distance criterion the winning neuron i(x) is found as follows:

$i(x)= \arg\min_j \|x-w_j\|, j=1,2,...,l$ (1) where l is the number of neurons in the grid.

4. Synaptic adaptation: after finding the winning neuron, all synaptic weights of each neuron vector are adjusted: $W_j(t+1)=W_j(t)+\eta(t)\Theta_j(t)(x(t)-W_j(t))$ where, t represents the current instant, $\eta(t)$ is the learning rate and $\Theta_j(t)$ is the neighborhood function.

5. Repeat steps 2 to 4 until no significant change happens in the topological map.

5 Conclusion

This paper has made a thorough analysis of identified security issues and the various solutions: architecture modification, algorithms and theorems that have been proposed to solve these issues. Though a relatively nascent research area in the investigation of SDN as a possible replacement of the existing network infrastructure, SDN security research effort has yielded ample success to support the assertion that SDN, in which the control plane is decoupled from the data plane is a better network architecture than the traditional network architecture and could serve as the network architecture of the future.

REFERENCES

[1] Krpeutz D., Fernando M.V.R, Paulo V., Christian E.R, Siamak A., Steve U., "Software-Defined Networking: A Comprehensive Survey." Proceedings of the IEEE, 2015 103(1): p. 14-76.

[2] Scott-Hayward, S., G. O'Callaghan, and S. Sezer. Sdn Security: A Survey. in Future Networks and Services (SDN4FNS), 2013 IEEE SDN for. 2013.

[3] Schehlmann, L., S. Abt, and H. Baier. Blessing or curse? Revisiting security aspects of Software-Defined Networking. in Network and Service Management (CNSM), 2014 10th International Conference on. 2014.

[4] Kloti, R., V. Kotronis, and P. Smith. OpenFlow: A security analysis. in Network Protocols (ICNP), 2013 21st IEEE International Conference on. 2013.

[5] Ashraf, J. and S. Latif. Handling intrusion and DDoS attacks in Software Defined Networks using machine learning techniques. in Software Engineering Conference (NSEC), 2014 National. 2014.

[6] Bing, W., Yao Z., Wenjing L., Thomas H., " DDoS Attack Protection in the Era of Cloud Computing and Software-Defined Networking". In Network Protocols (ICNP), 2014 IEEE 22nd International Conference on. 2014.

[7] Oktian, Y.E., L. SangGon, and L. Hoonjae. Mitigating Denial of Service (DoS) attacks in OpenFlow networks. In Information and Communication Technology Convergence (ICTC), 2014 International Conference on. 2014.

[8] Mowla, N.I., D. Inshil, and C. Kijoon. Multi-defense Mechanism against DDoS in SDN Based CDNi. in Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2014 Eighth International Conference on 2014.

[9] Zengguang, L., Y. Xiaochun, and L. Hoonjae. An Efficient Defense Scheme against SIP DoS Attack in SDN Using Cloud SFW. in Information Security (ASIA JCIS), 2014 Ninth Asia Joint Conference on. 2014.

[10] Lara, A. and B. Ramamurthy. OpenSec: A framework for

implementing security policies using OpenFlow. in Global Communications Conference (GLOBECOM), 2014 IEEE. 2014.

[11] Zaalouk, A., Rahamatullah K., Ronald M., Kpatcha B.,"OrchSec: An orchestrator-based architecture for enhancing network-security using Network Monitoring and SDN Control functions". In Network Operations and Management Symposium (NOMS), 2014 IEEE. 2014.

[12] Sousa, P., Alysson B., Miguel C., Nuno F., Paulo V.,Highly Available Intrusion-Tolerant Services with Proactive-Reactive Recovery. Parallel and Distributed Systems, IEEE Transactions on, 2010. 21(4): p. 452-465.

[13] Minzhe, G. and P. Bhattacharya. Controller Placement for Improving Resilience of Software-Defined Networks. in Networking and Distributed Computing (ICNDC), 2013 Fourth International Conference on. 2013.

[14] Phillip Porras, S.S., Vinod Yegneswaran† and M.T.Martin Fong, Guofei Gu, A Security Enforcement Kernel for OpenFlow Networks. 2012: p. 121-126.

[15] Zerrik, S., Amina O., Driss O., Rachid A., Mohamed B., Jaafa G., "Towards a decentralized and adaptive software-defined networking architecture. in Next Generation Networks and Services". (NGNS), 2014 Fifth International Conference on. 2014.

[16] Howard, m., Introduction to threat modeling.

[17] Shostack, S.Hernan and S.Lambert and T.Ostwald and A., modeling-uncover security design flaws using the stride approach, in MSDN. MSDN Magazine-Louisville.

[18] Dotcenko, S., A. Vladyko, and I. Letenko. A fuzzy logic-based information security management for software-defined networks. in Advanced Communication Technology (ICACT), 2014 16th International Conference on. 2014.

[19] Sezer, S., Sandra S., Pushpinder K., Barbara F., David L., Jim F., Marc M.,Navneet R., Neil V., "Are we ready for SDN? Implementation challenges for software-defined networks." Communications Magazine, IEEE, 2013.51(7): p.36-43.

[20] Beheshti, N. and Z. Ying. Fast failover for control traffic in Software-defined Networks. in Global Communications Conference (GLOBECOM), 2012 IEEE. 2012.

[21] FOUNDATION, O.N., SDN Security Considerations in the Data Center. ONF Solution Brief, 2013.

[22] Hu, Z., Wang M., Yan X., Yin Y., Luo Z., " A comprehensive security architecture for SDN. in Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on. 2015.

[23] Lamb, C.C. and G.L. Heileman. Towards robust trust in software defined networks. in Globecom Workshops (GC Wkshps), 2014. 2014.

[24] Braga, R., E. Mota, and A. Passito. Lightweight DDoS flooding attack detection using NOX/OpenFlow. in Local Computer Networks (LCN), 2010 IEEE 35th Conference on. 2010.

[25] Nguyen Tri, H.T. and K. Kyungbaek. Assessing the impact of resource attack in Software Defined Network. in Information Networking (ICOIN), 2015 International Conference on. 2015.

[26] Diego Kreutz, F.M.V.R., Paulo Verissimo Towards Secure and Dependable Software-Defined Networks, in HotSDN'13. 2013, ACM Hong Kong, China. p. 6.