SDN Issues – A Survey

Kapil Dhamecha
Department of Computer Science,
Rollwala computer center,
Gujarat University, Ahmedabad; India

Bhushan Trivedi, PhD.
GLS institute of computer tech,
Ahmedabad, Gujarat,
India

ABSTRACT:

SDN separates the control plane and data plane. SDN needs to be equipped with complex and proprietary networking devices as it needs to separate the infrastructure layer (network device) from the control layer (network OS, which provides a central view and control over the network and network services) and the application layer (software/business application)[1][6][8]. When a single controller is given the job of controlling multiple devices (switches/routers) these changes in network structure brought about by SDN will inevitably impact on network security. In this paper we studied issues and security challenges in SDN and current status.

Keywords:

Software Defined Network, Network Virtualization, Network security

1. INTRODUCTIONI

In the current scenario of networks, proprietary routers and switches firmware tells the network device where and how to forward the packets. Each network device (switch/router) has its own applications (routing protocols-OSPF, IS-IS), Network operating system and packet forwarding hardware make decisions solely on local logic as shown in fig 1. The switch task is sent every packet to the destination along the same path and treats all the packets the exact same way.

The current networks were built on the notion of Autonomous Systems (AS). This notion allows networks to scale and extend by connecting junctions that forward packets to a next hop based on information learned from the network and interconnection [9]. This process is easy and has proven sustainable, flexible and scalable for data networks. But the downside of current scenarios is AS principle does not allow the any designated destinations to move without changing their identity. This architecture relies on a treelike structure of Ethernet switches and routers. Initially VLAN technology confined to design network segment where virtual LAN controllers can change or add workstations, manage load balancing and bandwidth allocation without involving physical movement of nodes like conventional LANs. These standards on the other hand, have increased complexity in network element specifications and configuration of network interfaces by network operators. Thus such flexibility does not come without cost.

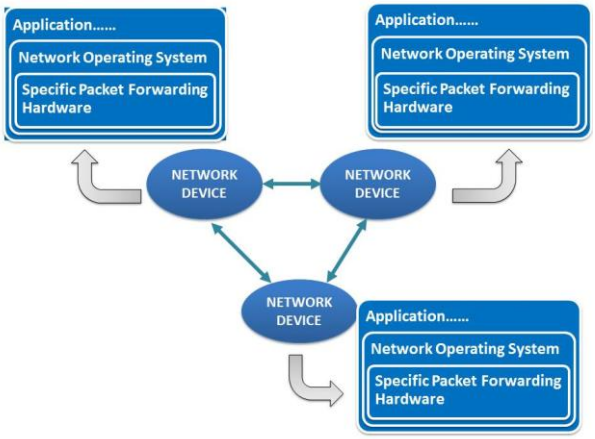

Fig 1 Current Network System

SDN architecture is depicted in figure 3. Control plane physically decoupled from the data forwarding plane and provide centralized control over the network. The controlling is logic may be run by a server which decides where and how to forward packet and data plane resides on network switch which forward packets by flow tables(similar to routing table and access control list(ACL))[3].

So SDN technology allows network operators to specify network services, without coupling these specifications with network interfaces, which will simplify extending VLANs (network segments) beyond the building perimeter, increasing the chances of data remaining secure and quickly changing to network requirements [8]. Essentially, this allows the administrator to use less expensive, commodity switches with more control over network traffic flow than ever before.

In SDN, network administrators can structure or shape the network traffic from centralized console, which introduces issues regarding availability and security of the network. Thus using SDN may result into disruption in a network which introduces changes in market dynamics. Small company wants to seize this opportunity to mark their spot in the industry [6] [10].

The current network is designed on specialize operating system, hardware and application which are designed by a particular vendor, which is closed propriety, vertically integrated and relatively slow innovation[8]. But SDN designed with open interface where everybody (network admin, security manager) able to use it and publish it, which introduce a system with horizontally integrated, open interface and rapid innovation. SDN also provides backward compatibility with Ethernet, IPV4, MPLS, and VLAN and construct a new and easy mechanism for forwarding with technology independent of switches and routers [10].

The rest of this article is organized as follows. Section 2 gives a brief description on basic SDN architecture and deployment model. In section 3 point out various SDN issues and security challenges in deployment of SDN. In section 4, evaluate and discuss various issues based on the analysis and suggest some solutions.

2. SDN ARCHITECTURE AND DEPLOYMENT MODELS

The upper layer of SDN architecture is an application layer which delivers various services such as firewall, intrusion detection system , metering, routing, QOS , load balancer etc.

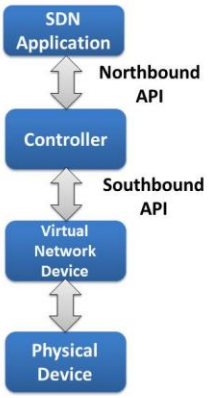

Fig 2 Basic SDN architecture

The 2nd upper layer consists of a controller which controls to prioritize, deprioritize, block specific level packets from traffic and also remotely control routing tables. There are many controllers available based on different programming language (python, c, c++, Java, Ruby , Javascript) platform such as POX, MUL, NOX, Jaxon, Trema, Beacon, Floodlight, Maestro, NDDI - OESS , Ryu, NodeFlow, ovs-controller etc.

The controller must provide a high degree of integration with lower and upper layer to provide communication between switches and application.

The 3rd upper layer consists of virtual network device which is nothing but a software emulation of physical switch. This switch function is switched, forwarding and scheduling of data traffic.

The lowest layer is providing infrastructure for IT. We use infrastructure as networking devices such as switches, router and you can also consider virtual part of devices as physical infrastructure.

As you have shown in fig 3, SDN API, it has two distinct API northbound and southbound.

The northbound API [8] provides an interface to the network operator and to the controller which can customize their network control on the fly, which can done by an average programmer using programming languages like c , c++ , python etc. It basically addresses for vendors and network service providers to customize applications themselves for in house or for academic use but till now there is no standard definition for northbound API. Each application will develop a view of the flow tables for network devices and then send requests to the controller for distribution to the network devices. The basic function of north bound API to include automatic management and sharing of data between systems.

The south bound API [8] provides mechanisms where end nodes (both physical and virtual switches/router) can talk to the controller via Openflow, OnePK API. So that network admin can discover network topology to define network flow and get a request relayed to it via north bound API to the application. Openflow used as a southbound application (figure 3).

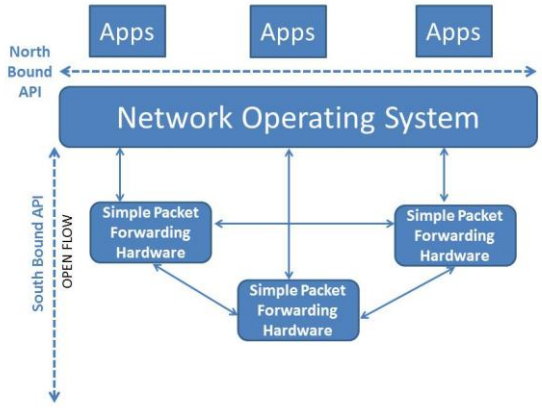

Fig 3 Software Defined Network System

The network administrator has complete control over network traffic through a software interface that SDN provides. This allows organizations to decrease their reliance on more expensive switches with proprietary firmware that performs these functions which must be set manually. They can do this from a centralized SDN allows network administrators to control network services more easily rather manually configured hardware. It also allows central programmable control of network without physical access to network hardware devices.

In the control mechanism, each of these switches/routers to SDN controller should be encrypted using SSL /TLS to provide an additional layer of security within the network.

There are two planes in network devices — a control plane that determines where traffic is sent and a data plane that forwards traffic based on what the control plane tells it to do. With SDN, these two planes have been detached (or decoupled) from each other. The data plane (*or* data forwarding plane) remains with the network hardware — but the control plane (or controller) that makes decisions about where traffic will be sent is now executed through software. This separation makes network Virtualization possible because you're no longer executing all the command or control rules on the hardware itself.

The SDN controller is used to configure the flow table within the remote switch/router. The controller use to make decisions based on the IP Address, Mac Address, TCP /UDP port Address and also can make statistics measure on traffic density. These addresses are defined in flow table which also use to defined rules, action and stats for specific flow. Open Flow is a protocol that uses APIs (application programming interfaces) to configure the switches in a network. SDN is software that gives network administrators a console interface where they can provision, manage, and break down networks without having to physically set up network switches and devices.

At the top-level, multiple control processes are talking to a single centralized controller, each communicating with controller for their purpose of work. Open Flow, as currently

specified, is not well-suited for use on top level, due to its lack of support for authorization and access control. But at the lower-level, a single centralized controller is used to control a group of switches/routers which are well-suited for communication between the centralized controller and the switches.

2. 1 various architecture models for SDN

2.1.1 Network Virtualization model [7]

Network Virtualization is an approach whereby several network instances can co-exist on a common physical network infrastructure. Network virtualization is solution to SDN. This is a simple model popularized by Nicira [12] to address the scalability issues with multicasting in Ethernet LAN based virtual network (VLAN) architectures. Network Virtualization platforms improvise a software element (hypervisor), but the cloud-building software (Open Stack) could also need to modify with an interface that creates VLANs that are based on network virtualized tunnels running on top of traditional Ethernet. Virtualization support both basic level of abstraction which is provided by physical network like L2, L3, ACL etc. are still supported in logic level of abstraction.

Advantage:

Network Virtualization still provides multi tenancy to cloud without need to modify it. [7]

Disadvantage:

It does not allow deep packet inspection to analyze packet header so the malicious data not detected by physical layer because the virtual layer is between the virtual network device not between users or devices.

2.1.2 Evolutionary model:

This model is used to enhance software control of the network and its operations but within the boundaries of current networking technology such as VLAN, AS, MPLS etc. And use them to partition the network into virtual communities and to manage traffic and Quality of Service. This model is fully integrated with network operations like fault management, configuration management, security management and also to traffic engineering principle can be applied.

Advantage:

As this model is software centric so the virtual networks can extend from server to user, as long as the devices support the selected standards.

Disadvantage:

Specific vendors offer evolutionary SDN models, which may not fully interoperate with equipment from other vendors.

2.1.3 Open flow model [6]:

Open Flow [6] is one such communication protocol that enables SDN. Open Flow, the first standard interface communications protocol designed specifically for SDN [8], decouples the control and data planes so that software can determine the network packets passing through a network thereby customizing the needs of applications at the application layer and its users. With the centralization of the control plane, it is possible to introduce and experiment with new capabilities in isolated slices of the network without affecting the rest of the network. This major change in network architecture offers its users a way to introduce new applications without the reliance upon individual device configuration and vendor releases. Using this model topology and application changes is reflected very quickly.

Advantage:

Make network operation simpler. It enables to share virtual and physical resource. It improves scale and performance of the network and it also provides control of the remote data plane of remote switch.

Disadvantage:

Version Compatibility issues with controller and OF switches.

There are different modes for Open Flow-based SDN networks which are:

i. Reactive [6]

In this mode, the first packet goes to switch/router, the packet is encapsulated, sent to controller the appropriate application interrogates it, make a decision how this packet forward in this network.

ii. Proactive [6]

In this mode, when the line is up, the controller goes to each individual switch/router and return with all the information to configure the flow table for packet forwarding. This architecture has the better idea about network topology, applications used in network and the best way to forward the traffic in the network.

2.2 Various SDN deployment models [16]

2.2.1 Symmetric vs. Asymmetric [16]

In Asymmetric model, a control mechanism is centralized and packet driving is distributed as much as possible so centralization makes easier consolidation and lowers the traffic aggregation. In symmetric model, control information about network is aggregate provide network reachability to every subset of a network.

2.2.2 Flood less vs. Flood based [16]

In flood based model, the global control information is either broadcast or multicast using to achieve a global view of the network topology and to find out the traffic density in specific subsets of the network. But the problem is created when any new location has added to this network will flood the network, which increase the traffic and reduce the scalability. In flood less model, all the forwarding is based on global exact match defined in the flow table, which is typically achieved using Distributed Hashing and Distributed Caching of SDN lookup tables.

2.2.3 Host based vs. Network centric based [16]

In host based model, basically this model use in virtual machine migration from host to host in large data center to provide elasticity .While during migration encapsulation of each VMs is limited to host only. So each host uses its own processing power very efficient to spare core capacity.

In the network centric model, clearly indicate the boundary line between the edge and end nodes. Such model more concentrated on the edges of the network rather count on end nodes for routing function.

3. ISSUES:

3.1 Network Virtualization issues [7]:

In a virtualized network, *network traffic does not pass through the physical network*, so there is no practical way to monitor data traffic in a virtual network. Security issues are related straightforwardly to virtualize networking because network monitoring application such as IDS/IPS, firewall, QOS services and access control list are useless at virtual layer. So such virtual machines will need additionally layer of security.

Similarly**,** *Isolation boundary* that controls the different virtual networks should be very well protected, If an attacker in one virtual network is able to detect the presence of other virtual networks, then they are *demolishing the illusion of separation [7]* in virtual network and also try to disrupt the whole network.

In programmable network , lack of well-structured policies and rules increase vulnerability in the network. So far, security issues that are specific to virtual networks are relatively unaddressed in the field. Specifically, the community has neither shown that virtual networks are as secure as traditional networks, nor provided enough security measures to defend them [12]. Hence, we expect this field to become increasingly important as network Virtualization technologies proliferate [7].

And other well-known security issues such as *confidentiality, integrity and availability*. Several solutions, such as authentication and intrusion detection have been designed to address such goals and to prevent attacks related to privacy, non-repudiation and man-in-the-middle [3].

Solutions:

SDN does not replace any existing security measures but the actually Virtual machine needs an additional layer as hypervisor security layer [7] to reduce attack in virtual network traffic and such solution taken into consideration that does not directly affect the flow data and flexibility provided by the SDN.

3.2 Controller:

3.2.1 Controller security issues:

When using a centralized SDN controller, *scalability and availability* challenges are present for both the control plane as well as the data plane.

At controller level *interoperability* is still an issue because no standardized platform till introduces for the controller.

Network intelligence is logically centralized in SDN controllers that maintain a global view of the network, which appears to applications and policy engines as a single, logical switch. So the majority of *networking security is evolve around controller is locked down*. So without proper security wrapped around it, the network becomes completely vulnerable to malicious attacks or unconfidently changes, both of which can take your network completely down.

Different applications may insert *different control policies dynamically* so how Open Flow controller guarantees that they are not conflicting with the other flow [5].

Open Flow controllers[3][5] do not uniformly capture and store *TCP session information* [1], among other key state tracking data, which is often required to develop security functionality [1] [4] [14] (e.g., TCP connection status, IP reputation).

When using TCP, it is recommended to use alternative security measures to prevent eavesdropping, controller impersonation or other attacks on the Open Flow SSL channel. [3]

There are many key questions arise while to think about centralized controls which are:

- Who is accessing the controller?
- The controller is available for the business continuity effort at valuable time?
- Is communication between controller and end nodes (switches/routers) is secured or not?

Available solution:

a) New application development framework ***FRESCO*** [4] - An Open Flow security application development framework designed to facilitate the rapid design and modular composition of Open Flow enabled detection and mitigation modules. It consists of an application layer and a security enforcement kernel, which are integrated into NOX open flow controller.

FRESCO's application layer is implemented using NOX [11] python modules, which are extended through FRESCO's APIs to provide two key developer functions: (i) a FRESCO Development Environment, and (ii) a Resource Controller [4]. It also provides interface to *FortNox* for better security at kernel layer and also offers several important features upon which FRESCO relies to ensure that flow rules derived from security services are prioritized and flow rules produced by non-security applications have less priority.

The authors also demonstrate FRESCO to introduce minimal overhead and its rapid creation of popular security functions with significantly (over 90%) fewer lines of code. [4]

b) ***FortNOX*** [5] is an extension to latest NOX [11] controller to provide three more roles based authentication for flow rule producer such as OF application role, Security role and operator role. The experimental security applications, and to more broadly understand the functional northbound requirements these applications need for detecting and responding to various attacks.

FortNox uses digital signatures to implement stronger application and rules provided by the producer. It uses alias set rule reduction for flow rule conflict on set operation (real time modification of flow) [5]. Network operator authorizes to define base security policies and apply these changes to the security application at any time.

FortNox uses conflict resolution policy to determine which rule should be accepted and which rule to be discarded. To reduce conflicts rules in flow FortNox consult digital signature of the conflicting flow rule to determine which rule associated with high authorization and which rule associated with low authorization. It also provides a logging function to record all activity of flow rule addition, deletion and updating.

c) ***Floodlight*** [9] –it is Apache licensed Java based Open flow Controller which provides an extension to security of Open flow protocol. Open flow based application provide an interface at both northbound and southbound side[6].

Extension to Open flow security to perform role based authentication, flow conflict detection for the controller.

3.2.2 Controller placement [2]:

SDN technology decoupled control planes which open many unanswered questions regarding availability, reliability, scalability, performance and network convergence time when compared to more traditional distributed systems [2]. The centralized control structure is more vulnerable, which needs proper work flow. The authors expected the answers depend on the desired reaction bounds, metric choice, and the network topology itself. More surprisingly, one controller location is often sufficient to meet existing reaction time requirements in network topology.

There are many questions arising regarding controller placement such as,

- How many controllers are needed?
- How does placement affect latency?
- Where in the topology should controller go?

The authors [2] analyzed controller placement in different 256 topologies which covers a diverse range of geographic areas (regional, continental, and global), network sizes (8 to 200 nodes), and topologies (line, ring, hub-and-spoke, tree, and mesh) this analysis provides some intuition for controller placement consideration in network topology. Reducing the average latency to half require three controllers, while the same reduction for worst-case latency requires four controllers for different topology scenarios.

Available solution:

Whenever an operator wants to add controllers, they should place controller such that network with minimum latency and response time. In many medium size networks, the latency from every node to a single controller can meet the response time goals of existing technologies and for large size network average latency is increase with response time, so it is feasible to use more than one controller in a distributed fashion.

3.3 SDN stacks (Figure 2):

Vulnerabilities also arise if the interaction between infrastructure layer and the control layer increased rather than an attack on the hosts or applications. Thus this may result into DDOS the SDN stack itself. i.e., Control Flow saturation attack [4].

Available solution:

Veriflow, it is a real time traffic debugger which is used to monitor malicious flow rules by the anomaly traffic detector and also prevent them reaching to the network.

4. CONCLUSION:

In this paper we have conducted a brief review of SDN most prominent security issues arise while deploying it in enterprise networks.

In table 1 (Appendix A) shows SDN current issues and their solutions. SDN still in starting phase of development, so many challenges arise while deploy it in current networks. As shown in section 3, some of the issues which related to Virtualization, controller placement/security and SDN stacks where some have reasonable solution, some have not. Virtual layer is abstracted from physical so L2/L3 packet header analysis not possible [7]. Other issues are related to centralized control which always addresses issues like availability, scalability and interoperability because till now no standards defined in controller [2]. Some of the issues are related to the integrity and confidentiality of flow which added by controller [3]. Controller level issues can be solved by monitoring the real traffic flow [1] coming to it from SDN applications such as IDS/IPS, logger tool and statistics measure techniques [3] [14] [1]. Veriflow find malicious flow rules on real time traffic but this affect the convergence time or speed of network [1].

The future direction includes introducing various security applications, insertion of extra virtual security layer above the physical layer for detail analyses of data traffic and introduce a new authentication mechanism to secure the controller.

5. REFERENCES:

[1] Khurshid, Ahmed, et al. "VeriFlow: verifying network-wide invariants in real time." Proceedings of the first workshop on Hot topics in software defined networks. ACM, 2012.

[2] Heller, Brandon, Rob Sherwood, and Nick McKeown. "The controller placement problem." Proceedings of the first workshop on Hot topics in software defined networks. ACM, 2012.

[3] Mehdi, Syed Akbar, Junaid Khalid, and Syed Ali Khayam. "Revisiting traffic anomaly detection using software defined networking." Recent Advances in Intrusion Detection. Springer Berlin Heidelberg, 2011.

[4] Shin, Seugwon, et al. "FRESCO: Modular Composable Security Services for Software-Defined Networks." To appear in the ISOC Network and Distributed System Security Symposium. 2013.

[5] Porras, Philip, et al. "A security enforcement kernel for OpenFlow networks." Proceedings of the first workshop on Hot topics in software defined networks. ACM, 2012.

[6] McKeown, Nick, et al. "OpenFlow: enabling innovation in campus networks." *ACM SIGCOMM Computer Communication Review* 38.2 (2008): 69-74.

[7] Wang, Anjing, et al. "Network Virtualization: Technologies, Perspectives, and Frontiers." *Journal of Lightwave Technology* 31.4 (2013): 523-537.

[8] Gurbani, Vijay K., et al. "Abstracting network state in Software Defined Networks (SDN) for rendezvous services." *Communications (ICC), 2012 IEEE International Conference on*. IEEE, 2012.

[9] Chanda, Abhishek, and Cedric Westphal. "Content as a network primitive." *arXiv preprint arXiv:1212.3341* (2012).

[10] Casado, Martin, et al. "Ethane: Taking control of the enterprise." *ACM SIGCOMM Computer Communication Review*. Vol. 37. No. 4. ACM, 2007.

[11] Gude, Natasha, et al. "NOX: towards an operating system for networks." *ACM SIGCOMM Computer Communication Review* 38.3 (2008): 105-110.

[12] Pfaff, Ben, et al. "Extending networking into the virtualization layer." *Proc. HotNets (October 2009)* (2009).

[13] Vaughan-Nichols, Steven J. "OpenFlow: The Next Generation of the Network?." *Computer* 44.8 (2011): 13-15.

[14] Braga, Rodrigo, Edjard Mota, and Alexandre Passito. "Lightweight DDoS flooding attack detection using NOX/OpenFlow." *Local Computer Networks (LCN), 2010 IEEE 35th Conference on.* IEEE, 2010.

[15] Elby, Stuart. "Software Defined Networks: A Carrier Perspective." *Proc. of Open Networking Summit* (2011).

[16] http://en.wikipedia.org/wiki/Software defined_networking.

[17] Handigol, Nikhil, et al. "Where is the debugger for my software-defined network?." *Proceedings of the first workshop on Hot topics in software defined networks*. ACM, 2012.

Appendix A:

Table 1- Comparison of SDN issue and their solution

Issues / **Different Problem category**	**Network virtualization issues[7]**	**Controller**		**SDN stacks issues**
		Security issues[3][4][5]	**Placement issues[2]**	
Prominent Reasons	Virtualized Operating system, Virtual network programming	Centralization of controller, no interoperability between standards ,flow authentication,	Depend on topology of network	Control flow saturate between controller and switches
Suggested/ available solution	Add extra hypervisor virtual security layer [7], design IDS/IPS, and firewall for this layer. Authentication and Access-Layer Security	Floodlight, FRESCO[4], FortNox	Controller may vary according to size of network[3]	Include Authentication or digital signature mechanism for higher priority given to flow
Available Resources	Node(NIC ,host/endpoint ,router) ,link (Bandwidth, tags/labels, Tunnels)	Controller platforms- NOX, POX, Floodlight, Jaxon, Trema ,beacon, Nodeflow ,MUL ,		N.A.
Affected parameter	Other Virtual Machines ,difficult to manage large amount of virtual machines state with consistency	Malicious flow	Latency, response time[2]	False flow rules[1]
Solution Provider	Xen, VMware[7]	Nicira, Bigswitch networks,		Veriflow[1],flowvisor , Ndb[17]
Open flow enable tool and application	NS3,Mininet,NICE-OF,Mirage,STS,Flowscale,ENVI,OFTest,Flowvisor, Route flow, Resonance ,SNAC			