International Journal of Electrical Electronics & Computer Science Engineering
Special Issue - ICSCAAIT-2018 | E-ISSN : 2348-2273 | P-ISSN : 2454-1222
Available Online at www.ijeecse.com

Strengthening Network Security: An SDN (Software Defined Networking) Approach

Pradeep Kumar Sharma[1], Dr. S. S. Tyagi[2]
[1]Ph.D Research Scholar, Computer Science & Engineering, MRIU, Faridabad, Haryana, India
[2]Professor, Computer Science & Engineering, MRIU, Faridabad, Haryana, India

Abstract: ***Today's IT infrastructure, featuring a mobile workforce, IoT applications, digital transformations of the business, and the cloud, is evolving at a pace that's exceeding the capabilities of legacy security approaches. Network security has proven to be insufficient, lacking the visibility, control, and intelligence necessary to keep up with changing needs. SDN has become a topic of interest in academia and industry as well to fulfill the on demand aspects of the network related to real time network policy enforcement. This paper evolutes the current security status and what can be done to improve it. In this paper we proposed an SDN based security approach which highlights the features which are not available in traditional security approaches. We show how the proposed security architecture can be used to strength the overall security of today's network.***

Keywords: SDN, Openflow, Control Plane, Data Plane.

I. INTRODUCTION

ICT infrastructures are expanding continuously with the increase in number of devices. At the same time, the anywhere, anytime mobile workforce, digital workplace transformation, IoT applications, and the move to the cloud are increasing the size and complexity of IT infrastructures and their associated attack surfaces. As a result network security is getting more complex every year. More on-premises and cloud applications, devices, users, and network traffic are increasing the attack surface, making it increasingly difficult to prevent, detect, and respond to security incidents. The concept of perimeter protection provided the old model for enterprise security. Today it takes more than a legacy firewall and antivirus software to protect your data. You need to look at the new innovative security solutions that are designed to address a world where the concept of a fixed perimeter is disappearing as the cloud becomes more pervasive.

Software defined networking is an emerging technology that can provide security defense solutions as it is capable of detecting attacks and acting adaptively in quicker way than traditional networks.[1]

The rest of the paper is as follows. Section II describes the problems in traditional security model and what is needed to improve. In section III we introduced SDN and Open flow. We present SDN based proposed security model for enterprise in section IV and we present related work in section V. Section VI concludes our analysis with summary and discussing the future work.

II. TRADITIONAL NETWORK SECURITY AND PROBLEMS

With the increase in IT infrastructures, cloud computing and increase in number of devices, network security and network management[2] is getting more complex every year. Organizations today confronting with a world of ever-evolving security threats and there is a little choice but to rely on a combination of security solutions that are complicated, distributed, and limited in scope. Security policies are typically implemented as complex, topology-dependent access control lists. Trust is distributed across multiple components, such as switches, Domain Name System (DNS) servers, and authentication services (such as Kerberos, and Remote Authentication Dial In User Service (RADIUS); and each of these individual components need to be protected in turn. If a network element is compromised, an attacker may be able to identify vulnerabilities and obtain sensitive information about the network itself, such as the topology, the location of critical servers, etc.

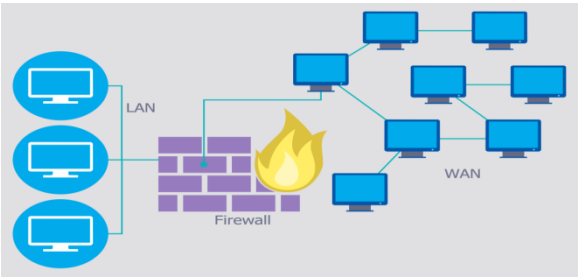

Fig. 1. Traditional Network Security

Figure1 and 2 shows the traditional network architecture where the security has been implanted at the edge and Organizations and their infrastructures are left exposed to a dangerous threat landscape where persistent cyber-attackers have proven adept at bypassing aging security mechanisms. In figure 3 we have shown the SDN architecture of network security and we have compared how the architecture in figure 3 is different from figure 2. With the SDN security architecture the malicious traffic can be redirected and blocked in real time based on the traffic patterns.

International Journal of Electrical Electronics & Computer Science Engineering
Special Issue - ICSCAAIT-2018 | E-ISSN : 2348-2273 | P-ISSN : 2454-1222
Available Online at www.ijeecse.com

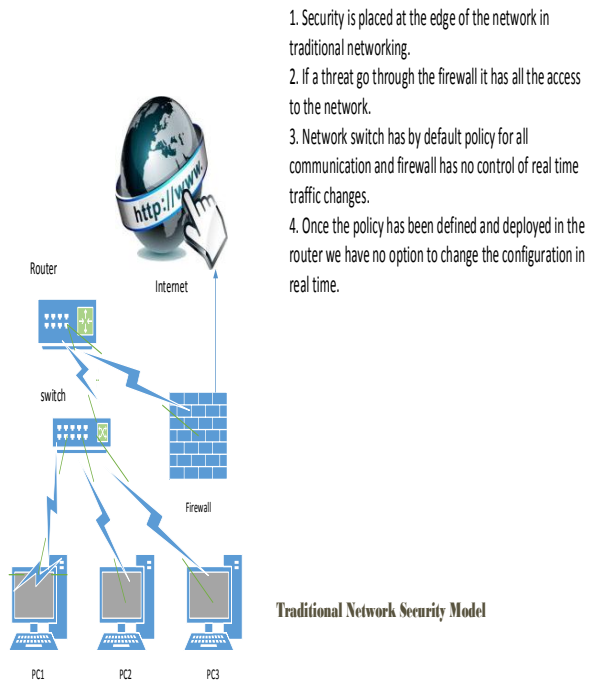

Fig. 2. Traditional Network Security Model

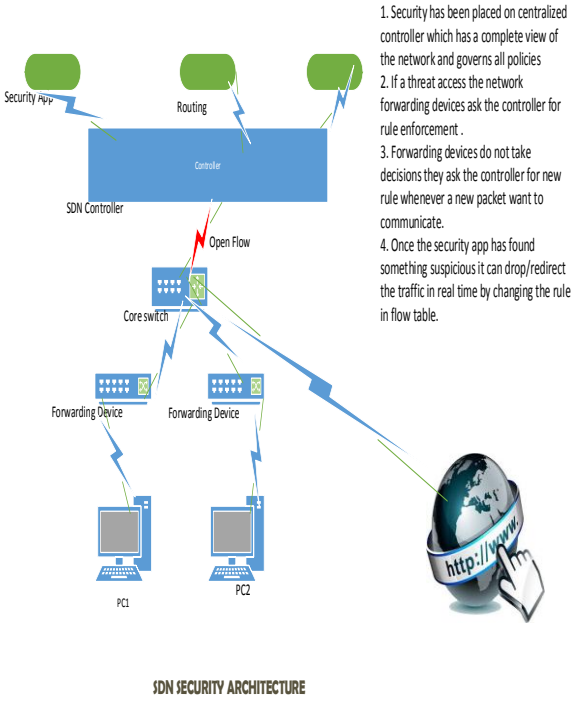

Fig 3. Proposed Security with SDN

III. SOFTWARE DEFINED NETWORKING AND OPENFLOW

To counter the above complexity of the traditional network infrastructure SDN is a new paradigm which divides the network functionality into control and data plane. In SDN controlling part of the network element has been decoupled from the data plane and a centralized controller is used for controlling and configuring the network devices.

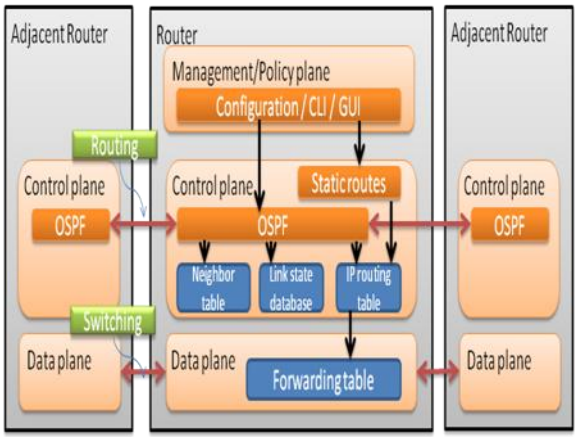

Fig. 4. Control and Data Plane in Router

Figure 4 Shows the control and Data plane in a networking device (e.g router).

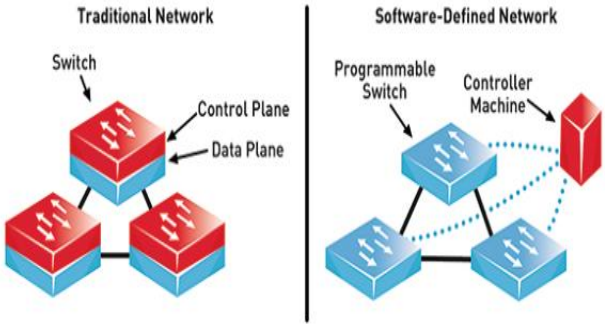

Fig. 5. Separation of Control and Data Plane

The control plane controls the decisions and configuration while data plane perform the data forwarding. Figure 3 shows the separation of control and data plane in SDN. For clarity, SDN is described in this article with the Open Networking Foundation (ONF) [3] definition: "In the SDN architecture, the control and data planes are decoupled, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications."

SDN focuses on four key features:

- Separation of the control plane from the data plane
- A centralized controller and view of the network

International Journal of Electrical Electronics & Computer Science Engineering
Special Issue - ICSCAAIT-2018 | E-ISSN : 2348-2273 | P-ISSN : 2454-1222
Available Online at www.ijeecse.com

- Open interfaces between the devices in the control plane (controllers) and those in the data plane
- Programmability of the network by external applications
- Increased Security and reliabilities with complete visibility and control over the network

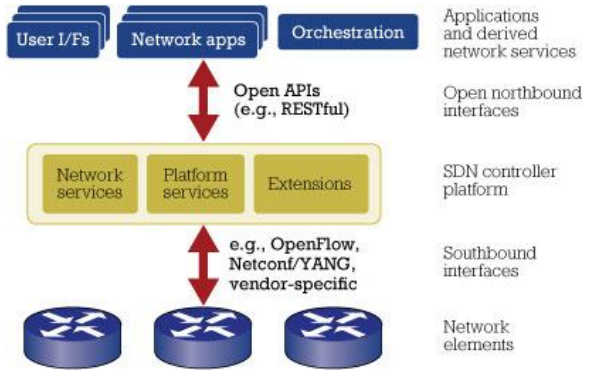

Fig. 6. SDN Architecture

In figure 4 we can see the SDN architecture where network elements i.e forwarding devices communicate with SDN controllers with the help of openflow protocol. OpenFlow[4] is a protocol which defines the flow rules between controller and data plane where switches in the network can be reduced to basic packet forwarding devices containing flow tables populated with localized flow rules. These rules describe how incoming packets will be handled. The rules are managed remotely by a controller entity, which communicates securely with switches potentially using a standard and open interface, such as Open Flow Protocol. To reconfigure a switch to enable a new policy, the controller modifies the relevant entry in the flow tables and this modification may also be done reactively; for example, a specific packet may arrive at the switch and controller updates the existing flow rules or specifies new ones accordingly in real-time.

Currently most research work focuses on exploiting SDN to improve network security[5] by using monitoring systems with inserting security policies to dynamically detect and mitigate suspicious traffic during live network operations. Mehdi et al. [6] propose the value of using SDN to provide intrusion detection in a Home Office/Small Office environment. OpenSAFE [7] uses its ALARMS policy language to manage the routing of traffic through network monitoring devices. A Distributed DoS (DDoS) detection method based on several traffic flow features to identify abnormal flows is presented in [8]. CloudWatcher [9] controls network flows to guarantee that all necessary network packets are inspected by some security devices with introducing SDN in the cloud. This framework automatically detours network packets to be inspected by pre-installed network security devices. FleXam [10] enables the controller to access packet-level information for the detection and mitigation of botnet and worm propagation. FRESCO [11] presents an OpenFlow-enabled security application development framework designed to facilitate the rapid design and development of security modular for the detection and mitigation of network threats.

IV. SDN BASED PROPOSED SECURITY MODEL

In SDN logical centralization[12] of network intelligence and complete view of the network presents exciting challenges and opportunities to enhance security in networks, including new ways to prevent, detect, and react to threats, as well as innovative security services and applications that are built upon SDN capabilities. Packet filtering can be done at different levels of the network and most of them are able to analyze the packet headers up to the transport layer. Achieving traffic filtering above layer 4 can be enabled by OpenFlow by inspecting the packets at controller side assuming that all incoming packets are forwarded to it with PacketIn messages.

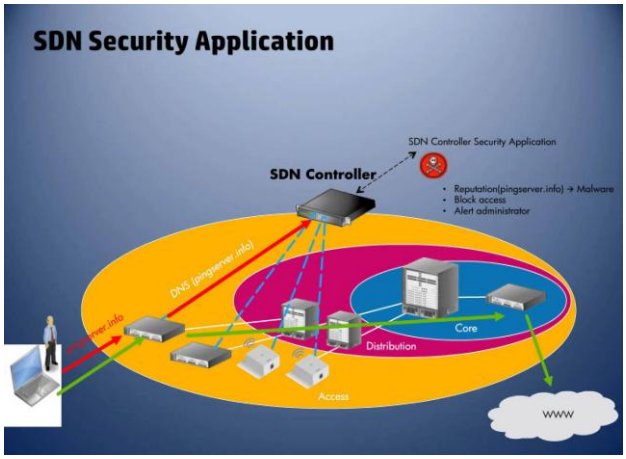

Fig. 7. SDN Security Application Scenario

The architecture shown in the figure has been depicted for the purpose of presentation and identification of the tools and components necessary to validate the research proposal and related data. To perform the simulations we have chosen mininet. Mininet is a SDN network emulator. It runs a collection of end-hosts, switches, routers, and links on a single Linux kernel. It uses lightweight virtualization to make a single system look like a complete network, running the same kernel, system, and user code. A Mininet host behaves just like a real machine; you can ssh into it (if you start up sshd and bridge the network to your host) and run arbitrary programs. The programs you run can send packets through what seems like a real Ethernet interface, with a

International Journal of Electrical Electronics & Computer Science Engineering
Special Issue - ICSCAAIT-2018 | E-ISSN : 2348-2273 | P-ISSN : 2454-1222
Available Online at www.ijeecse.com

given link speed and delay. Packets get processed by what looks like a real Ethernet switch, router, or middlebox, with a given amount of queueing. When two programs, like an iperf client and server, communicate through Mininet, the measured performance should match that of two (slower) native machines.

A. Model Solution Implementation Control Plan:

In mininet we used POX as an SDN controller which is a framework based on components for programming software for Software Defined Networks, which uses python as a programming language and allows you to program applications using multiple protocols e.g OpenFlow protocols[13]. In mininet framework we are provided with a hub code we check the behavior of HUB in SDN environment and we convert this hub to openflow support switch using python code. Again we check the behavior of the switch with SDN controller and hosts.

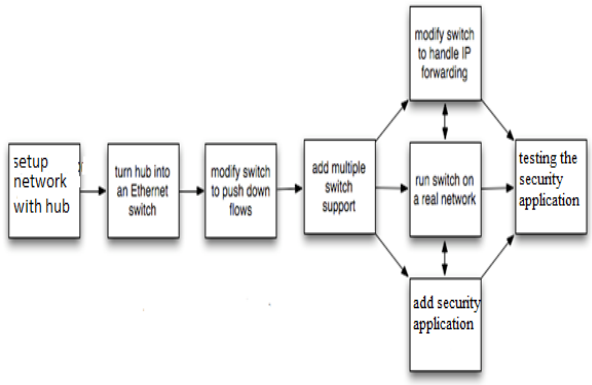

Fig. 8. Work Flow of the Proposed Model

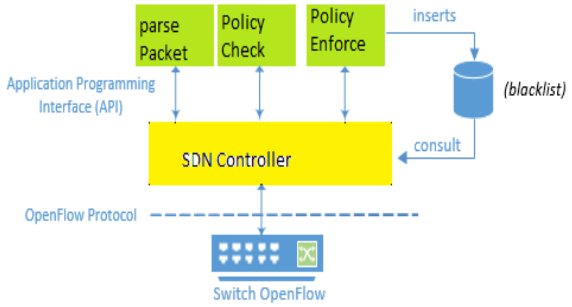

Fig. 9. SDN Security Application Building Blocks

Figure shows the implementation scenario of SDN based network security proposed application with SDN/Openflow. When a switch receives a packet it check the flow table, if the related entry is found in flow table the packet is sent to the destination. If no rule is found in the switch flow table the packet is sent to the controller and controller sent the packet to security application for security policy checking. The security application first parses the received packet, checks whether the incoming packet violates the security policies or not and enforces a flow rule based on the security policies. Finally this rule is delivered to switch by the controller and switch update the rule in its flow table. Packet is blocked based on some event associated with an attack signature in the openflow network through Packet_event messages and further packets from this sender blacklisted by the security application.

B. Description of Safe Proposed Network Architecture:

Figure shows the network architecture of the proposed setup. We used an openflow switch s1 which is connected with three hosts h2, h3 and h4 having IP address 10.0.0.2, 10.0.0.3 and 10.0.0.4 respectively. The switch is connected with controller C0, with loopback address 127.0.0.1 on port 6633.

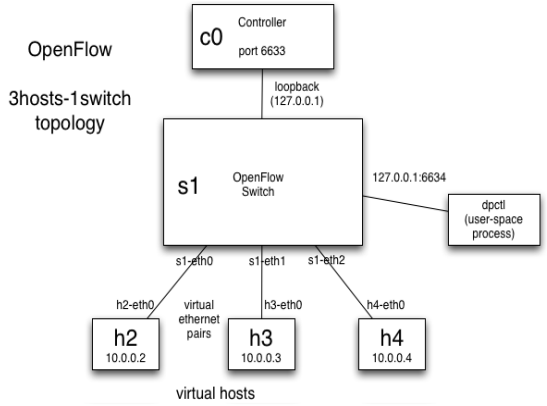

Fig. 10. Simulation Setup

The simulation setup consists of following components:

- VirtualBox console terminal: connects to OpenFlow image. This is the one created when you started up the VM.
- SSH terminal: connects to OpenFlowTutorial. Created by using putty on Windows or SSH on OS X / Linux.
- xterm terminal: connects to a host in the virtual network. Created in the next section when you start up the virtual network. Will be labeled at the top with the name of the host.
- OpenFlow Controller: sits above the OpenFlow interface. The OpenFlow reference distribution includes a controller that acts as an Ethernet learning switch in combination with an OpenFlow switch. You'll run it and look at messages being sent. Then, in the next section, you'll write our own controller on top of NOX or Beacon (platforms for writing controller applications).

- OpenFlow Switch: sits below the OpenFlow interface. The OpenFlow reference distribution includes a user-space software switch. Open vSwitch is another software but kernel-based switch, while there is a number of hardware switches available from Broadcom (Stanford Indigo release), HP, NEC, and others.
- ovs-ofctl: command-line utility that sends quick OpenFlow messages, useful for viewing switch port and flow stats or manually inserting flow entries.
- Wireshark: general (non-OF-specific) graphical utility for viewing packets. The OpenFlow reference distribution includes a Wireshark dissector, which parses OpenFlow messages sent to the OpenFlow default port (6633) in a conveniently readable way.
- iperf: general command-line utility for testing the speed of a single TCP connection.
- Mininet: network emulation platform. Mininet creates a virtual OpenFlow network - controller, switches, hosts, and links - on a single real or virtual machine.
- cbench: utility for testing the flow setup rate of OpenFlow controllers.

V. CONCLUSIONS

The challenge of implementing security strategy based on the paradigm with SDN openflow is the focus of the proposed security architecture. The technique, now associated with network control mechanism of SDN/openflow allows an innovative way not only to detect the threats as well to react the threats in a controlled manner. This setup includes a single switch and 3 host, as a future work a more elaborated version of this architecture can be implemented and tested with more than one switch with load balancing and filtering the malicious traffic.

VI. REFERENCES

[1] Syed Taha Ali et. Al., "A Survey of Securing Networks using SDN" 'IEEE transactions on reliability, Vol 64, No. 3, Sep 2015.

[2] P.K Sharma, S.S Tyagi, " Simulation of an SNMP Agent:Operations, Analysis and Results", IJECSE, Vol.1, no. 4, pp.1919-1927, 2012.

[3] ONF, "Software-Defined Networking: The New Norm for Networks," white paper, https://www.opennetworking.org

[4] N. McKeown, T. Anderson, H. Balakrishnan, G.Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," ACM SIGCOMM Comput. Commun. Rev. (CCR), vol. 38, no. 2, pp. 69–74, 2008.

[5] K. Tomar, S.S Tyagi, "HTTP Packet Inspection Policy for Improvising Internal Network Security", IJCNIS, Vol. 6, no. 11, pp.35-42, 2014. DOI: 10.5815/ijcnis.2014.11.05

[6] S. A. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting traffic anomaly detection using software defined networking," in Recent Advances in Intrusion Detection. Springer, 2011, pp. 161-180.

[7] J. R. Ballard, I. Rae, and A. Akella, "Extensible and scalable network monitoring using opensafe," Proc.INM/WREN, 2010.

[8] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in IEEE 35th Conference on Local Computer Networks (LCN). IEEE, 2010, pp. 408-415.

[9] S. Shin and G. Gu, "CloudWatcher: Network security monitoring using OpenFlow in dynamic cloud networks (or: How to provide security monitoring as a service in clouds?)," in 20th IEEE International Conference on Network Protocols (ICNP). IEEE, 2012, pp. 1-6.

[10] S. Shirali-Shahreza and Y. Ganjali, "Efficient Implementation of Security Applications in OpenFlow Controller with FleXam", in 21st IEEE Annual Symposium on High-Performance Interconnects. IEEE, 2013, pp. 49-54.

[11] S. Shin, P. Porras, V. Yegneswaran, M. Fong, G. Gu, and M. Tyson, "FRESCO: Modular composable security services for software-defined networks," in Proceedings of Network and Distributed Security Symposium, 2013.

[12] Marc. C. Dacier et. Al. " Security Challenges and Opportunities of Software Defined Networking ", in IEEE Computer and Reliabilities Societies, pp.96-100, March 2017.

[13] Sin-Fu Lai et. Al. " Design and Implementation of Cloud Security Defense System with Software Defined Networking Technologies" , IEEE, pp.292-297, 2016.