Enhancing Network Security: an SDN (Software Defined Networking) Method

A. Nandhini, Mohamed Althaf, K. Priyankha, F. Femila

Abstract: ***Today's IT infrastructure, featuring a mobile workforce, IoT apps, digital business transitions and the cloud, is developing at a rate that exceeds traditional technology approaches capabilities. Network security has proven insufficient and lacks the visibility, control and intelligence necessary to meet changing needs. SDN became a subject of interest for academia and industry to discuss the network on-demand relevant aspects of applying real time network policy. The article is an assessment the security situation and what to do to strengthen it. In this paper we introduced SDN-based approach functionality not available in traditional approaches to security. This show how the proposed security architecture can be used fortify the overall protection of today's network.***

Keywords: Control Plane, Data Plane, Open-flow, SDN

I. INTRODUCTION

With the number of devices growing, ICT infrastructures are constantlyexpanding.

At the same time, IoT systems, mobile employees, digital w orkplace transformation, and cloud changes all over the place increase the size and complexity of IT infrastructures and their related attack surfaces. More on-site and cloud apps, devices, users and network traffic are that the surface of the attack, making it harder to avoid, track and respond to security incidents. As a result, the security of the network becomes more complex each year.

The perimeter defence framework established the old enterprise security model.

What it takes today to protect your data is more than just a legacy firewall and antivirus software. The new, innovative security solution needs to be looked at designed to address an environment where a fixed perimeter paradigm is vanishing as the cloud becomes more omnipresent. Software defined networking is an emerging technology that can provide security defence solutions because it can detect attacks and act in a faster fashion than traditional networks.

Section II discusses issues in the conventional security model, and what needs improvement. We implemented the SDN and Open flows in section III.

In section IV, we present proposed company security model focused on SDN and current related work in section V. Section VI completes the study by summing up and considering future work.

Revised Manuscript Received on March 30, 2020.
***** Correspondence Author

Nandhini Alagarsamy*, Department of CSE, Sri Krishna College of Technology, Coimbatore, Tamil Nadu, India.
E-mail:16tucs125@skct.edu.in
Priyankha Kailasam, Department of CSE, Sri Krishna College of Technology, Coimbatore, Tamil Nadu, India.
E-mail:16tucs144@skct.edu.in
Mohamed Althaf, Department of CSE, Sri Krishna College of Technology, Coimbatore, Tamil Nadu, India.
E-mail:16tucs116@skct.edu.in
Femila F., Assistant Professor, Sri Krishna College of Technology, Coimbatore, Tamil Nadu, India.

II.CONVENTIONAL NETWORK SECURITY AND PROBLEMS

Cloud computing and increasing number of computers, network security and network management are becoming more complicated each year with the rise in IT infrastructures. Organizations today face a world of ever-evolving security threats and there's a little option but to rely on a mix of complex, centralized, and scope-limited security solutions. If a function in the network is breached, vulnerabilities can be identified by an attacker and sensitive network information such as topology, key server location, etc.

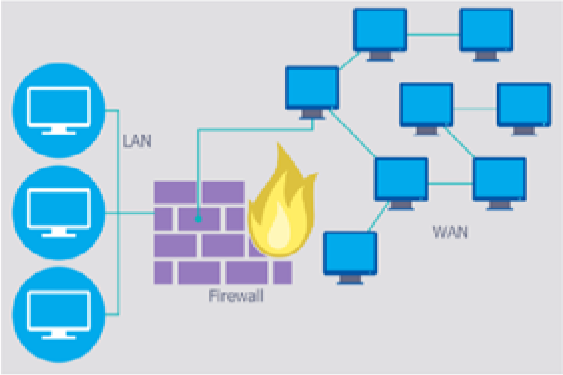
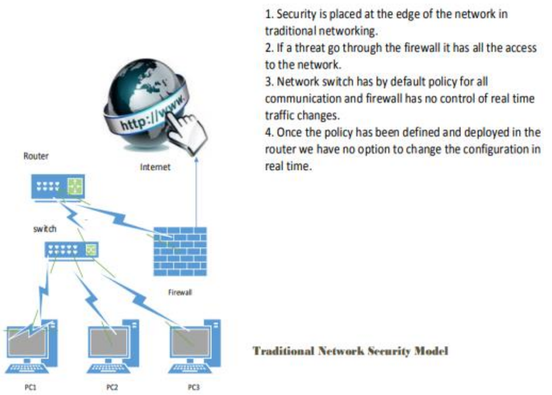


Fig.1 Conventional Network

Figures 1 and 2 show the conventional network architecture in which protection was developed at the edge, companies and their networks were exposed to a risky world in which persistent cyber attacks were able to circumvent outdated security mechanisms. The SDN architecture for network security was implemented in Figure 3 and in Figure 3 we compared how the architecture differs from Figure 2.

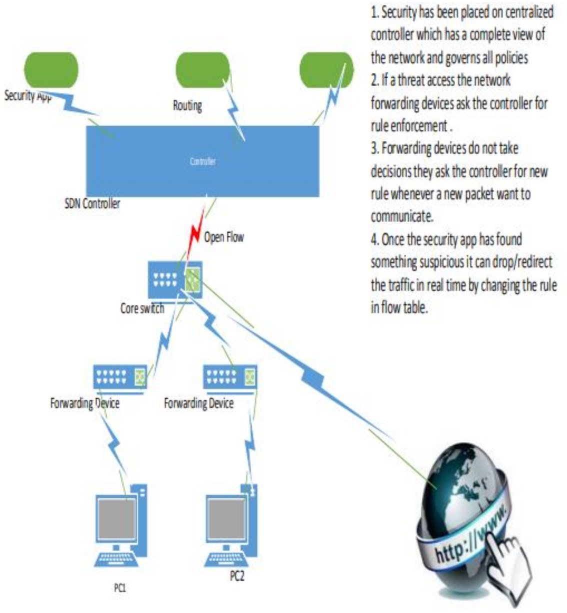


1. Security is placed at the edge of the network in traditional networking.
2. If a threat go through the firewall it has all the access to the network.
3. Network switch has by default policy for all communication and firewall has no control of real time traffic changes.
4. Once the policy has been defined and deployed in the router we have no option to change the configuration in real time.

Traditional Network Security Model

Fig.2. Conventional Network Security Model

DOI: 10.35940/ijitee.E2143.039520



1. Security has been placed on centralized controller which has a complete view of the network and governs all policies
2. If a threat access the network forwarding devices ask the controller for rule enforcement .
3. Forwarding devices do not take decisions they ask the controller for new rule whenever a new packet want to communicate.
4. Once the security app has found something suspicious it can drop/redirect the traffic in real time by changing the rule in flow table.

Fig 3.Suggested protection with SDN

III. NETWORK SUPPORT AND APPLICATE

Combating the alluded to above complexities of the conventional SDN network infrastructure is a new paradigm that separates Command and Data Plane features of the network. The network devices are manage and configured using SDN data plane and a centralized controller.The portion of the network feature management was removed from the SDN data plane, and a centralized controller is used to track and configure network devices.

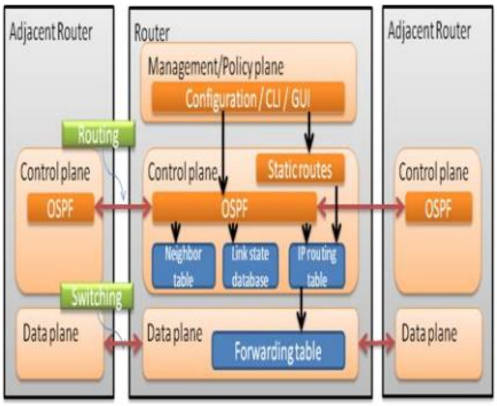


Fig. 4. Command and Route Data Plane

Figure 4 defines a networking device's control and data plane (e.g. router).

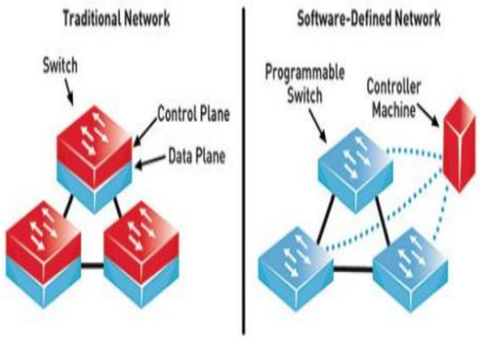


Fig. 5. Command and Data Plane Divide

Figure 3 demonstrates how SDN divides power and data plane. For clarity, definition of SDN as base for Open Network (ONF) is described in this article: "Command and Route data planes are decoupled in the SDN architecture state are logically centralised, and the applications are abstracted from the underlying network infrastructure".

SDN focuses on four key features: ·

- Command plane isolation from data Plane.
- Open interfaces between controlplane devices (controllers) and data plane devices.
- Network programming through global technologies.
- Better protection and stability with full view and monitoring of the network.

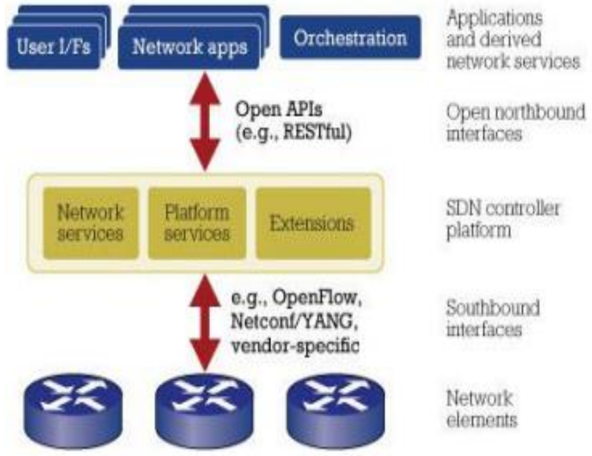

User I/Fs
Network apps
Orchestration
Applications and derived network services
Open APIs (e.g., RESTful)
Open northbound interfaces
Network services
Platform services
Extensions
SDN controller platform
e.g., OpenFlow, Netconf/YANG, vendor-specific
Southbound interfaces
Network elements

Fig. 6. Architectural style with a SDN

Figure 4 illustrates the SDN architecture where, using Open Flow protocol, network elements.

The rules are centrally managed by a controller agency which, like the Transparent Flow Protocol, can communicate securely with switches through a regular, transparent interface.

Many inquiries is currently focused on leveraging SDN to enhance network security by using security policy monitoring systems to automatically identify and minimize traffic suspicious during live network service. In it is given a Distributed DoS (DDoS) detection system based on several traffic flow characteristics. Cloud Watcher tracks network flow to ensure that some security devices check all relevant network packets when integrating SDN into the cloud. FRESCO offers a security application development platform provided by Open Flow.

IV. SDN BASED PROPOSIT MODEL SECURITY

The strategic centralization of intelligent networks and the full networking view in SDN offers exciting challenges and opportunities to enhance network security, including new ways of preventing, monitoring, and reacting to attacks, as well as creative security services and SDN centric applications. Packet filtering can be performed at various network rates and most should be able to test the packet headers up to the transport layer.OpenFlow will allow the filtering of traffic above layer 4 by inspecting the controller side packets if all incoming packets are sent with Packet-In messages.

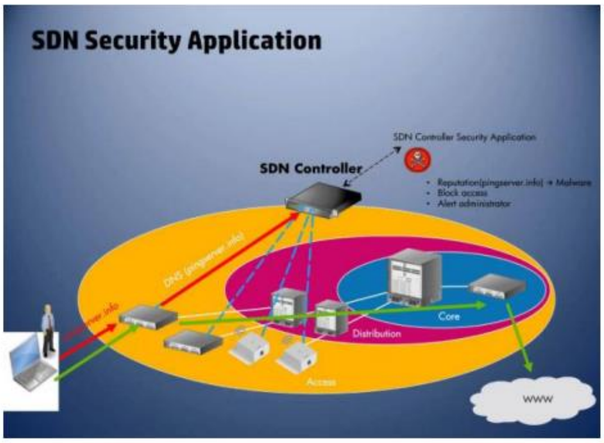


Fig.7.SDN Protection Demand Scenpicture

To present and clarify the tools and components needed to validate the proposal for research and the related information. Mininet is emulator for the SDN network. On a single Linux kernel it operates a collection of end-hosts, switches, routers, and links. A single system looks like a complete network using Lightweight virtualization, running the same kernel, applications and usercode. A Mininet host operates just like a real computer; if you start sshd and link the network to your server, you can ssh it in and run arbitrary programs. The programs you run will send packets, with the speed and delay of a given link, through what appears to be a real Ethernet interface. Packets are handled by what appears to be a rea Ethernet switch, modem, or middle box, with some queue. If two programs communicate via Mininet.

A.Plan for Implementation of System Solution:

We used POX in mininet as a SDN checker which is a component-based programming system for Software Defined Networks, allowing you to program applications using multiple protocols.In the SDN environment, we test HUB's behavior and use python code to convert the hub to OpenFlow support switch.

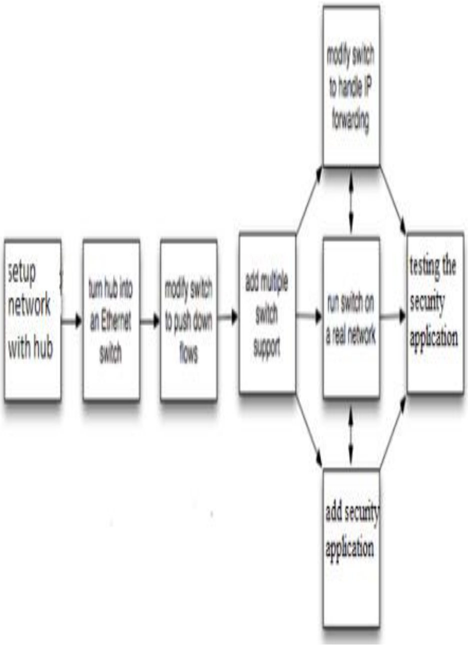

setup network with hub
turn hub into an Ethernet switch
modify switch to push down flows
add multiple switch support
modify switch to handle IP forwarding
run switch on a real network
add security application
testing the security application

Fig. 8. Proposed platform workflow

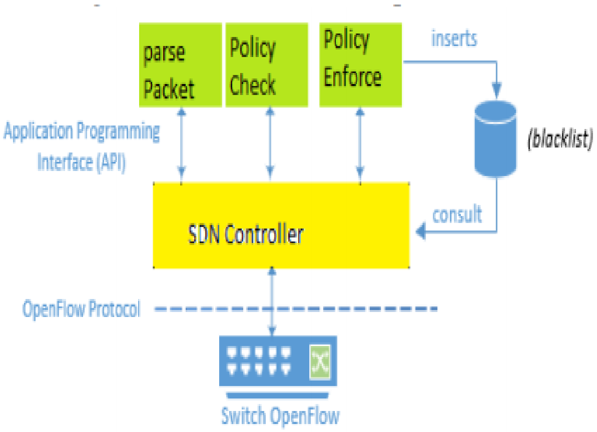


Fig.9.SDN Protection Demand Building Blocks

Figure shows the scenario for implementation of the proposed framework with SDN / OpenFlow for the SDN based network security. If a switch receives a packet, if the related entry is included in the flow table, the flow table will be checked. Eventually, the controller uses this rule in their flow chart to transform and change the rule.

B. Secure Infrastructure Proposed Network Description:

The figure shows the architecture of network proposed for the setup. We used an Open Flow switch s1 attached to IP address 10.0.0.2, 10.0.0.3 and 10.0.0.4 with three host h2, h3 and h4. The switch is attached to controller C0, with loopback address 127.0.0.1 in port 6633.

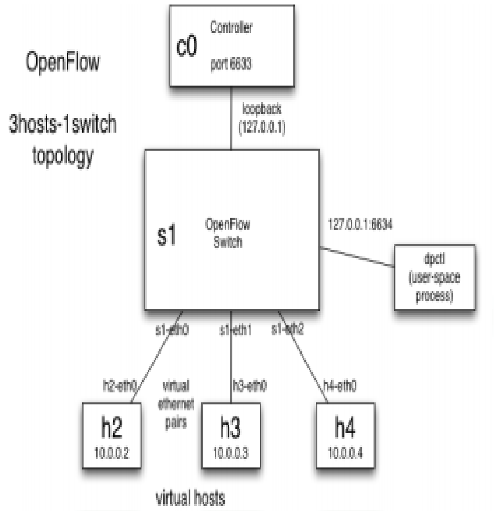


Fig.10. Configuration of simulation

The Simulator System consists of the following elements:

- VirtualBox Remote Console:
  Links to OpenFlow image. You created this one t o start the VM.
- SSH: Logs into OpenFlow Tutorial.
- OpenFlow checker:Lies over OpenFlow controller. The OpenFlow reference implementation contains a controller,which combines to serve as an Ethernet learning switch with an OpenFlow switch. You will then On top of NOX or Beacon write our own controller in the next section.

- OpenFlow Switch: is located under the OpenFlow gui. The reference distribution for OpenFlow includes a switch for the space consumer program.
  Open vSwitch is another software but a kernelbased switch, while a range of hardware switches are available from
  Broadcom , HP, NEC, and others.
- Ovsofctl: command line utility that sends short OpenFlow messages, Can be used to show port switches and flow information, or to manually insert flow inputs.
- iperf: general command line utility to check a single TCP connection speed.
- Mininet: Architecture for emulation of networks. Mininet establishes a virtual OpenFlow network on a single real or virtual computer-controllers, switches, hosts, and connections.
- Cbench: Tool for testing OpenFlow controller flow setup rates.

V. RESULT

New Features Provided by SDN SDN/OpenFlow provides programmability, dynamicity, flexibility, and intelligence to current network architectures, and its benefits can be delivered from four main features: (i) dynamic flow control, (ii) network-wide visibility with centralized control, (iii) network programmability, and (iv) simplified data plane. Dynamic Flow Control: Based on SDN's basic characteristics (i.e., ask the control plane if the data plane does not have a flow rule to handle a network flow), a network application can control network flows dynamically. This feature is highlighted with network applications for flow control, such as dynamic load balancing and network management application. Network-Wide Visibility with Centralized Control: In SDN, all data planes are connected to a centralized control plane to receive control messages (e.g., flow rule insertion and data plane configuration). In addition, the control plane collects network status information from each data plane by sending a statistics query message. Therefore, a network application running on the control plane naturally has a view of all connected data plane, and it can control all data plane in a centralized way. Several network-wide monitoring applications with SDN are good examples that benefit from this feature. Network Programmability: Since all data planes in an SDN network can be controlled by a network application program, SDN provides a strong capability to program enable new network functions. This is similar to programming a smartphone (e.g., Android) app to enable unlimited creativity of functionalities. To empower this feature, several network programming languages have been proposed so far, and they help us program network functions easily. Simplified Data Plane: Basically, the SDN architecture separates the data plane from the control plane, and thus the data plane only has relatively simple logic .This simplified data plane gives us chances of adding some new features NetFPGA ,DevoFlow are good examples of the simplified data plane and its modification.

TABLE I. OVERALL SUMMARY OF SDN FEATURES

SDN Feature	Feature Description	Benefit to Security	Network/Security Application Examples	Roles in Defense
Dynamic flow control	SDN can control (e.g., reroute, forward, drop) network flows dynamically	Dynamically control malicious or suspicious network flows (packets), separate malicious network flows from benign flows	FlowVisor, OpenVirtex, FlowN, splendid, NVP[30], Random route mutation, Random host mutation, Varmour, FlowNAC, PBS	Prevention, Response
Network-wide visibility with centralized control	All network status and flow information can be monitored and managed by a centralized server, which we call a controller	Monitor whole network in a centralized way for security services, detect network flooding or network anomaly efficiently and effectively (network-wide monitoring)	CloudWatcher, NetSecVisor, SIMPLE, FlowTags, OpenNF, SPHINX, DDoS detection/defense, Resonance, NetFuse, FleXam	Detection, Response
Network programmability	SDN enables us to program network functions	Develop network security applications easily, open the gate of devising advanced network security applications	FRESCO, Nettle, Frenetic, OpenSAFE, Procera, Controller Programming	Detection, Response
Simplified data plane	SDN makes the data plane quite simple by moving out complicated control plane logic	Change the data plane light weightly as a kind of security device by adding new modules	Avant-Guard, OFX, OpenSDWN	Prevention, Detection, Response

VI. CONCLUSION

At the heart of the proposed security architecture lies the challenge of implementing a security strategy based on the SDN OpenFlow model. The method, now connected to the SDN / OpenFlow network control mechanism, not only helps to identify treads in a creative manner but also help to respond to threats in a controlled manner.

This design requires a single switch and 3 host, and it is possible to implement and check a more complicated version of this architecture with more than one switch with load balancing and malicious traffic filtering as future work.

REFERENCE

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.
14. Pradeep Kumar Sharma and S.S.Tyagi “Strengthening Network Security: An SDN (Software Defined Networking) Approach”, ICSCAAIT-2018 | E-ISSN : 2348-2273 | P-ISSN : 2454-1222

AUTHORS PROFILE

Nandhini Alagarsamy, IV CSE, Sri Krishna College of Technology, Coimbatore, Tamil Nadu, India. Email:16tucs125@skct.edu.in

Priyankha Kailasam, IV CSE, Sri Krishna College of Technology, Coimbatore, Tamil Nadu, India. Email:16tucs144@skct.edu.in

Mohamed Althaf, IV CSE, Sri Krishna College of Technology, Coimbatore, Tamil Nadu, India. Email:16tucs116@skct.edu.in

Femila. F, Working as an assistant professor at Sri Krishna College of Technology, Coimbatore. She received her B.E in Computer Science Engineering from Jeppiar Maamallan Institute of Technology, Chennai, Tamil Nadu, India in the year 2010 and M.E in Computer Science Engineering from SKR Engineering College, , Chennai, Tamil Nadu, India in the year 2012. Her area of interest Data Science.