SECURITY OF SOFTWARE DEFINED NETWORK WITH SOFTWARE DEFINED PERIMETER

Co-authored by Zeal Egaesiri Ekrebe

Pavol Zavarsky

Sergey Butakov

Project report

Submitted to the Faculty of Graduate Studies,

Concordia University of Edmonton

in Partial Fulfillment of the

Requirements for the

Final Research Project for the Degree

MASTER OF INFORMATION SYSTEMS SECURITY MANAGEMENT

Concordia University of Edmonton

FACULTY OF GRADUATE STUDIES

Edmonton, Alberta

April 2020

SECURITY OF SOFTWARE DEFINED NETWORK WITH SOFTWARE DEFINED PERIMETER

Zeal Egaesiri Ekrebe

Approved:

Pavol Zavarsky [Original Approval on File]

Pavol Zavarsky Date: April 14, 2020

Primary Supervisor

Edgar Schmidt [Original Approval on File]

Edgar Schmidt, DSocSci Date: April 16, 2020

Dean, Faculty of Graduate Studies

Security of Software Defined Network with Software Defined Perimeter

Zeal Ekrebe, Pavol Zavarsky, Sergey Butakov
Information System Security and Assurance Management
Concordia University of Edmonton
Edmonton, Alberta, Canada
zekrebe@csa.concordia.ab.ca, pavol.zavarsky@concordia.ab.ca, sergey.butakov@concordia.ab

***Abstract*—There has been notable proliferation of network-enabled devices. The myriad of security threats and highly publicized data breaches demonstrate the importance of protecting access to the evolving enterprise network. The introduction of Software Defined Network (SDN) has brought about remarkable improvements to network performance, scalability, administration (in terms of flexibility and management). Software Defined Perimeter (SDP) has also some remarkable improvement to network security. This paper examines the pros and cons of a possible integration of SDN and SDP – investigating the potential of an improved set-up, effectiveness, management, and of course security of today's fast growing proliferation of networks. While carrying out the research and in consultation with referenced papers, this paper; (1) examined possible security threats and/or vulnerabilities that may be exploitable in SDNs, (2) examined Software Defined Perimeter and what it could mean for the security of entire network architecture (Software Defined Network), and (3) investigated the possibility of integrating SDN and SDP into one solution (on the premise that what is unknown or unaccepted cannot be harnessed).**

Keywords— Software Defined Network (SDN), Software Defined Perimeter (SDP), Mininet, POX Controller.

I. INTRODUCTION

It has become very apparent, essentially undeniable, that technology has become and is a very important piece in businesses of today. The footprint of technology is clearly seen from the use of as simple as a cell phone/ smart-phone; to the use of IP phones in offices and business structures; to the use of tablets, laptops and desktop computers; to the use of highly technologically driven machines (Artificial Intelligence for example) used in business processes and production. This involvement of technology leads to the transmission, as well as processing of information and its elements for both a given business and her customers. With this necessary collection and transfer of information, comes lots of concerns, risks and threats closely associated with the proliferation of enterprise networks of today. Recently, there has been a notable amount of buzz around Software Defined Networking as an option to carter for these proliferation needs. This paper examines the pros and cons of a possible integration of SDN and SDP – investigating the potential of an improved set-up, effectiveness, management, and of course security of today's fast growing proliferation of networks.

SDP is an approach to cyber security that mitigates network-based attacks by ensuring that all connections to services available and running on a network infrastructure are secure. SDP uses a need-to-know model in which device posture and identity are verified before access to application infrastructure is granted; whether the assets are on-premise or in a public or private cloud, a DMZ, a server in a data centre, or even inside an application server.

SDN is an approach to computer networking that gives network administrators the ability to manage network services through abstraction of higher-level functionality – a separation of the control plane from the data plane. SDPs and SDNs both essentially have same architectural setup – control plane (controller), a flow protocol (for communication between the control and data planes), and a data plane (where the devices are found); making an integration possible.

The contributions of this paper can be summarised as follows;

1. This paper assumed some worse-case scenarios that may pose security threats and/or vulnerabilities that may be exploitable in SDNs, based on what we know of SDNs.

2. This paper also examined Software Defined Perimeter and resulting implication on the security of entire network architecture (Software Defined Network) in comparison to what is obtainable with Traditional Perimeter Model for security.

3. This paper also investigated the possibility of having a single controller with the ability to set-up/ design a network (SDN), and the ability to provide security for the network (SDP). We experimented with POX controller in Mininet.

II. RELATED WORKS

A. Definition of Concepts

a) SDN: Paper [1] defined SDN as a new model that can elastically expand or shrink with the dynamic change in a network without affecting the network's overall performance, by separating the control plane from the data/ forwarding plane. Paper [1] further divided the SDN network into three main levels; the data plane, the control plane, and the application plane. While talking about the control plane, paper [1] said the plane is managed by the SDN controller with basic functions including, flow table management, link discovery, topology management, strategy making, storage management, and control data management; consequently allowing for the addition of many applications and features as needed. While talking about the data plane, paper [1] said that the plane consists of a set of forwarding devices which are commonly known as SDN switches, although they may not contain the basic function of layer 2 switches. Paper [2] said that SDN abstracts the forwarding plane from the control and

management plane. Paper [2] said also said that the separation of the control and forwarding plane, provides a scalable distributed solution. In paper [4], Software Defined Networking (SDN) is defined as a networking paradigm in which the control and management of the network is separated from the traffic forwarding primitives.

b) SDP: Papers [1], [9], [20], [25] & [26] defined SDP as a solution to secure the application network infrastructure based on a need-to-know model, in which device identity is verified before granting access. Paper [20] further said that SDP ensures security whether the assets are on-premise or in a public or private cloud, a DMZ, a server in a data centre, or even inside an application server. Paper [25] further said that SDP has been proposed as a security model/ framework to protect modern networks in a dynamic manner.

B. Problems and Solutions

While talking about the separation of the control plane and the data plane in SDNs, paper [1] said that the separation opens security challenges categorised into two levels; Outer level, and Inner level. The Outer level services are exposed to/ targeted for Denial of Service (DoS) and Distributed Denial of Service (DDoS) attack, aimed at exposing dedicated server resources and/ or to slow legitimate users causing a DoS [1]. In the Inner level of SDN systems, the centralised controller presents a potential single point of failure that is vulnerable to network manipulation; allowing an intruder to possibly compromise the SDN controller and/or one of the SDN applications, produce false network data, and initiate different attacks on the network [1]. Paper [1] proposed the use of SDP for security solution which addresses the three tiers of security – Confidentiality, Integrity, and Availability. Paper [1] said that the SDP workflow goes through several layers of security which gives protection while patching most vulnerabilities found in legacy security systems. Firstly, the gateway firewall in SDP contains a static drop-all policy allowing SDP to effectively repel flooding attacks [1]. Secondly, the use of Single Packet Authorisation (SPA) mitigates attacks by allowing the server to discard a DoS attempt before entering the TCP handshake [1][25]. Thirdly, the connection between all hosts (Initiating Hosts and Accepting Hosts) must use Transport Layer Security (TLS) or Internet Key Exchange (IKE) with mutual authentication to validate the client as a legitimate member of the SDP prior to furthering device validation and/ or user authentication [1]. All of these layers are capable of preventing network manipulation attacks, MiTM attacks, and packet/ traffic sniffing attacks. In fact, the Cloud Security Alliance (CSA) SDP Hackathon challenged hackers to attack a server defended by SDP; of the billions of packets fired at the server, not one attacker penetrated even the first layer of security [1].

Paper [4], [19] & [22] discussed detailed overview of exploitable SDN vulnerabilities. Paper [4], after discussing some SDN vulnerabilities, proffered a solution with the use of SDN security evaluation methodology developed to serve as a guideline for evaluating security implementations in an SDN infrastructure affecting choice of SDN controller to use in a given infrastructure. In paper [19], focus was duly given to all aspects of security – Confidentiality, Integrity and Availability; but there was no solution given to the said possible vulnerabilities.

Paper [15] focused on one aspect of the security triad (availability) – DoS, resulting from limitations of the capacity of a single and centralized controller. Paper [15] proposed a dynamic load balancing mechanism for distributed controllers based on a hierarchical architecture.

Paper [16] & [18] focused on one aspect of the security triad (availability) – DoS, resulting from possible link congestion as a result of a centralised controller. Paper [16] proposed an effective routing mechanism in SDN that can mitigate link congestion; utilizing a topology detection module, monitoring module, and traffic rerouting module. Paper [18] proposed the use of an algorithm in programming the network to mitigate congestion; the algorithm works on minimizing the load on a single link by choosing alternate paths of same length that will eventually increase the throughput of the system and also extend the lifetime of the various elements in the network design.

Paper [17] focused on one aspect of the security triad (availability) – DoS attacks on the SDN controller. Paper [17] proposed an integrated approach of attack trees and extension innovation methods from the perspective of attack identification for security management of SDN.

Paper [26] while discussing some challenges and concerns of modern networks, divided the said challenges into three main categorises; (1) Security Challenges/ Concerns (primarily Confidentiality in the CIA triad), (2) Privacy Challenges/ Concerns (primarily Integrity in the CIA triad), and (3) Availability Challenges/ Concerns (primarily availability in the CIA triad). Paper [26] further said that SDP is a potential solution to tackle many of the security and privacy challenges facing future networks [26].

C. Focus Papers

Paper [1] & [2] both proposed an integration of SDNs and SDPs (though paper [2] uses the terms “SDSec” or “SDSecurity”) but with different approaches from themselves and even in relation to our work.

The work in [1] proposed an integration of SDN and SDP that required implementations of SDN and SDP as standalone solutions linked by a Switch; with the goal of protecting a set of services connected to an SDN network by embedding the SDP components into the SDN environment [1]. The paper focused on scaling in a virtualized environment; as well as the latency and throughput of both solutions (SDN and SDP solutions) and their impact on the test-bed used. This idea is really admirable as it takes

care of single point of failure attacks (a challenge faced in a single centralised SDN controller implementation). This approach however, brings about independent manageability and control of the independent systems which could inherently pose some challenges to the network administration. And as stated in section VI of paper [1], although their "proposed architecture can protect the SDN Outer level, more challenges still exist to secure the Inner level, mainly the SDN Controller"(the security challenges in the Inner level is mitigated by the use of Identity-based Access in our experiment). Paper [1], similar to this paper, is concerned with all three aspects of the CIA triad – Confidentiality, Integrity, and Availability as can be seen in the two scenarios they (authors of paper [1]) used for their experiment; Client-to-Gateway and Client-to-Server.

The work in [2] is closer to work done in this paper in terms of its use of one controller in its implementation. It, however, can be differentiated from work done in this paper, with its focus on virtualised environment only (this is not a demerit, just a little different from the approach used in this paper). The nomenclature paper [2] chose – "SDSecurity" (Software Defined Security) seems to be well suited for the proposed integration of SDN and SDP. The proposed solution in paper [2] does however, seem to be just a security solution and not a solution that handles design of a network, as well as security of the network (once again, not a demerit, just a technical difference from the approach used in this paper). Paper [2] in its introduction said; "The new technology (SDSecurity) provides a new way to design, deploy and manage security mechanisms by separating the forwarding and processing plane from the security control plane, in a similar way as SDN abstracts the forwarding plane from the control and management plane." It may also present a single point of failure that may not be found in the proposed implementation of [1]. Paper [2] proposed SDSecurity experimental framework/ platform which is built upon Mininet by customising and extending the elements of Mininet to facilitate building a virtualised environment to emulate the different forms of SDSec policies and test their performance under different scenarios. The major difference of paper [2] from work done in this paper is that, their (authors of paper [2]) research essentially focuses on availability (in the CIA triad) – DoS attacks and the reduced impact the targeted device/ resource experiences when their "SDSecurity DoS attack detection" is used, in comparison to when their "SDSecurity DoS attack detection" is not used.

Most of the related works addressed the issue of DoS and its impact on availability (just one aspect of security; leaving out Confidentiality and Integrity – other two aspects of the security triad); proposing algorithms to carryout load-balancing; few also proposed a distributed system infrastructure as opposed to having a centralised system architecture (one centralised controller for example). Some other related works focused on exploring and giving detailed reports on vulnerabilities that might be exploited with the use of SDNs without proffering possible mitigations/ solutions to the vulnerabilities. Still unanswered, therefore, is the possibility of integrating SDN and SDP into one holistic solution enabling the design, centralised manageability, and of course security of a network; not as standalones as proposed in paper [1], or building an alternative to SDP (SDSec) that focuses on availability (DoS attacks) as proposed in paper [2].

III. CONTRIBUTIONS EXPLAINED

A. Objective 1: Possible security threats and/or vulnerabilities that may be exploitable in SDN

At the moment there are no obvious attacks on SDN implementations, maybe due to the fact that not many have adopted the SDN approach to networking. However, this may not be the case in the event that SDNs become more commonly placed in fast-paced, ever growing networks of today. Hence, we examined the need for security of Software Defined Networks – resulting in a clear realisation of security – Confidentiality, Integrity, and Availability for SDNs. So what are some security implications - threats and/or vulnerabilities that may be exploitable in SDNs?

With focus on the Controller (which sits in the Control Plane) some possible attacks could be:

a) Compromise of the Controller: For very obvious reasons the controller would be a major target for attackers as it is "the seat of power".

- The objective may be a DoS attack, due to compromised integrity of the software components of the controller which may essentially cripple the controller and the network at large.
- Another objective, similar to the first, but in this case capitalizing on the known vulnerabilities of the operating system on which the controller runs. For example, some controllers run on some model of Linux OS. The vulnerabilities of the OS could inherently put the controller at risk. This risk might pose a challenge to the integration of SDN and SDP as well.

b) Introduction of a rogue Controller: A rogue controller may also be introduced to the network, which could be very damaging, depending on the intent of the attacker.

With focus on the Controller Network Elements (which sits in the Data Plane) some possible attacks could be;

c) Attack from within the Network: An attack could spring from within the network.

- An attacker having unauthorized access either physically or virtually to the network could pose a threat.
- Introduction of an already compromised host can pose serious threats from within the network.
- Insider attack from a privileged user or administrator.

d) Utilizing loopholes/ vulnerabilities of South-bound APIs and Protocols: Protocols and APIs used by the Controller to communicate with the Network Elements, if poorly implemented, may present vulnerabilities that could be exploited. For example, an attacker can try to initiate traffic flows – spoofing flows to permit traffic types that may have been denied by the administrator of the network; worse yet, steer traffic in his (attacker) direction – constituting possibly eavesdropping/ MiTM (Man in the Middle) attack.

With focus on the Applications running on the network (which sits in the Application Plane) some possible attacks could be;

e) Utilizing loopholes/ vulnerabilities of North-bound APIs: North-bound APIs used by the Controller may present vulnerabilities that could be exploited. Attacks may be geared or targeted towards the controller's management interface – maybe essentially sabotaging the configuration of the SDN environment.

A more detailed overview of exploitable SDN vulnerabilities can be seen in [19] & [22]. The above mentioned security implications - threats and/or vulnerabilities that may be exploitable in SDNs and the ones in [19] & [22], bring to the fore, the need for security of Software Defined Networks; this is where the Software Defined Perimeter comes into play.

B. Objective 2: Software Defined Perimeter and resulting implication on the security of entire network architecture (Software Defined Network) in comparison to what is obtainable with Traditional Perimeter Model for security

a) Software Defined Perimeter (SDP): SDP is an approach to computer security which evolved from the work done at the Defence Information Systems Agency (DISA) under the Global Information Grid (GIG) Black Core Network initiative around 2007, and now being adopted by the Cloud Security Alliance for its membership [9][20][25][26]. SDP ensures that all connections to services available and running on the networking infrastructure are secure, based on a need-to-know model in which device posture and identity are verified before access to application infrastructure is granted [9][20][25][26]; whether the assets are on-premise or in a public or private cloud, a DMZ, a server in a data centre, or even inside an application server [20]. SDP's ability to ensure security of connections, not just cloud based connections, is what makes the SDP concept a valuable solution to the security of SDNs. Due to its selective process, the SDP infrastructure is referred to as "black"; because it cannot be detected by users who are unauthorised to see it [25][26].

In Objective 1, while discussing some possible threats and vulnerabilities, we mentioned threats from vulnerabilities in the layering OS on which the SDN may have been built on. We did conclude by saying that the risk might pose a challenge to the proposed integration of SDN and SDP as well. Worthy of note however, is that the SDP's workflow through several layers of security, gives maximum protection to the systems [1]. SDP does offer a level of obscurity of the infrastructure that should help to mitigate issues having to deal with vulnerabilities inherently found in the underlying SDN OS. Also noteworthy are the following; (1) gateway firewall in SDP contains a static drop-all policy allowing SDP to effectively repel flooding and PS attacks [1], (2) the Secure authentication feature in SDP mitigates attacks by allowing the server to discard the DoS attempt before entering the TCP handshake [1][2], (3) the connection between all hosts must use TLS or Internet Key Exchange (IKE) with mutual authentication to validate the client as a legitimate member of the SDP prior to furthering device validation and/or user authentication [1]. All of these layers are capable of preventing network manipulation attacks, MiTM attacks, and traffic sniffing attacks. The Cloud Security Alliance (CSA) SDP Hackathon challenged hackers to attack a server defended by SDP. Of the billions of packets fired at the server, not one attacker penetrated even the first layer of security [1]. All of these SDP features would help in mitigating risk associated with OS vulnerabilities.

b) Changing Perimeter from Traditional Model to Software Defined Perimeter Model: Historically, enterprises deployed a perimeter security solution in their data center to protect against external threats to their application infrastructure. However, the traditional perimeter model is rapidly becoming obsolete for two reasons; (1) Hackers can easily gain access to devices inside the perimeter (for example via phishing attacks through BYOD devices) and attack application infrastructure from within. Moreover, this vulnerability continues to increase as the number of devices inside the perimeter grows due to Bring Your Own Device (BYOD), on-site contractors, and partners. (2) Traditional data centre infrastructure is being supplemented with external resources such as PaaS, IaaS, and SaaS. Subsequently, networking equipment used for perimeter security is topologically ill-located to protect application infrastructure. The growth of devices moving inside the perimeter and the migration of application resources to outside the perimeter has stretched the traditional security model used by enterprises. Existing workaround solutions that involve backhauling users to a data centre for identity verification and packet inspection do not scale well [19].

SDP on the other hand, combines and integrates (1) device authentication, (2) identity-based access, and (3) dynamically provisioned connectivity to hide critical applications from hackers [20][25].

This research paper in no way belittles the beautiful and reliable traditional security model that has been leveraged and depended on over the years for the security of network infrastructure. However, SDP seems to be better suited as a promising solution to protect against the security challenges/ concerns facing modern networks – especially when an infrastructure is designed with placement on site, as well as on the cloud [25][26].

C. Objective 3: The possibility of integrating SDN and SDP into one solution

Investigating the possibility of having a single controller with the ability to set-up/ design a network (SDN), and the ability to provide security for the network (SDP), we experimented with POX controller in Mininet.

Since SDPs and SDNs both essentially have same architectural setup – control plane (controller), a flow protocol (for communication between the control and data planes), and a data plane (where the devices are found); an integration could possibly look like the representation in Figure 1.

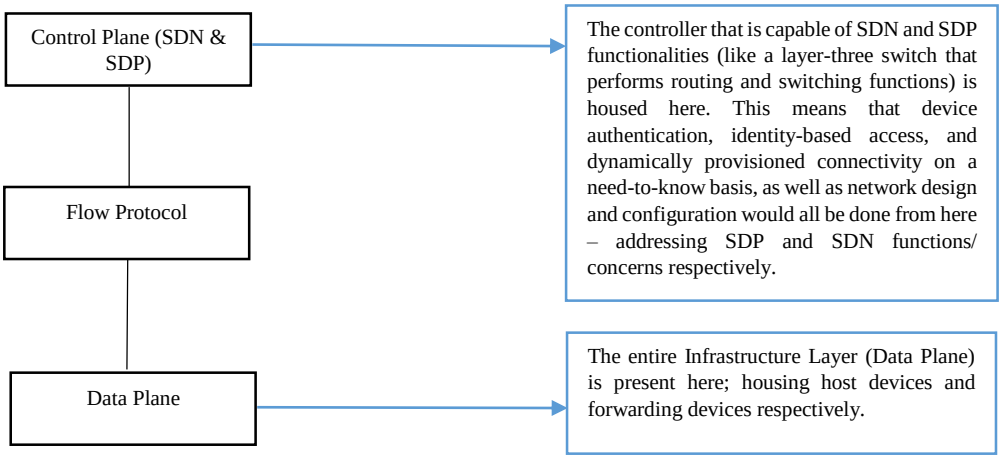

Figure 1. An example of what integration of Software Defined Perimeter and Software Defined Network would look like architecturally.

Realistically, the proposed integration is very similar to what is seen in layer-three switches. Ordinarily a switch is a layer two device, which simply means that it does not perform routing operations. However with layer-three switches, routing operations are possible alongside switching capabilities. All that is done is an improvement, more like an added function; the basic architecture stays the same. The experimental work done in this paper with the POX Controller in Mininet provides credence thereof.

Table 1 shows SDP and SDN (POX controller) components/ functions mapping from a security standpoint.

TABLE 1. SDP AND SDN (POX CONTROLLER) COMPONENTS/ FUNCTIONS MAPPING FROM A SECURITY PERSPECTIVE

No.	SDP Sec. Features	SDN (POX controller) Sec. Features
1.	Identity-based Access	misc.firewall.py component
2.	Device Authentication	misc.firewall.py component; misc.mac_blocker component

IV. EXPERIMENTAL WORK AND RESULTS

This paper chose to work with POX Controller in Mininet. Mininet is simply an open-source tool that helps in achieving network emulation. The reason for choosing to use the POX controller is the fact that it is designed with some security features - making it more than just a SDN controller; it is also open-source making it easily accessible for research purposes.

A network topology was created having a controller – POX Controller (control plane), a switch and five host devices (data plane). Two key scenarios were used: scenario 1, to demonstrate Identity-based Access (SDP feature); scenario 2, to demonstrate Device Authentication provisioning (SDP feature). Figure 2 shows the topology used in experiment. The experimental network was built using Mininet.

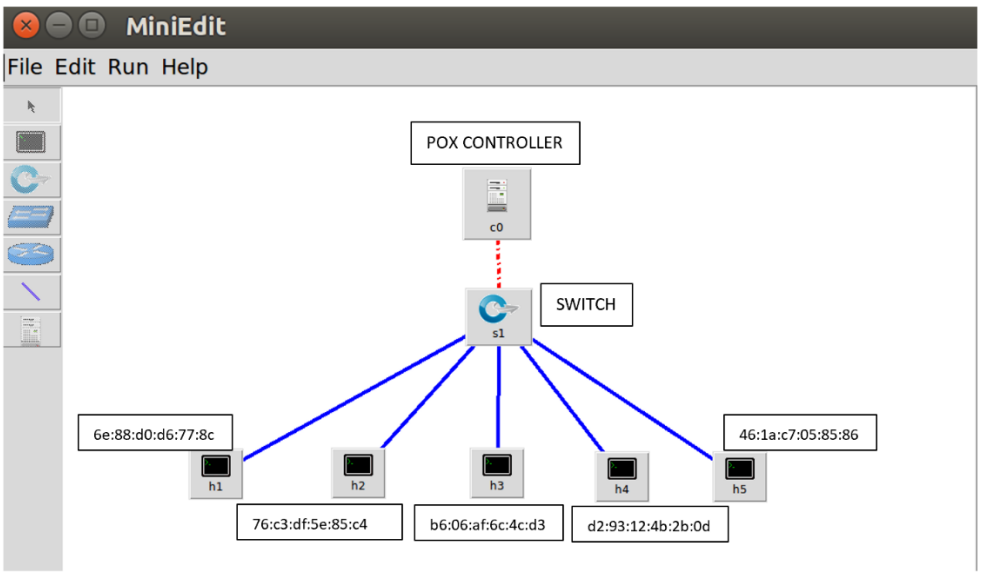

Figure 2. Topology used to conduct experiment.

A. *Scenario 1*

In this scenario, device identity-based access using *misc.firewall* component in the POX Controller was simulated. The said component was used to implement traffic restrictions in the network based on the identities of the network's hosts. Assuming that H1 to H5 are possibly working in different departments, or different security clearance levels; therefore, they would have different access levels based on their identities.

```
mininet> pingall
*** Ping: testing ping reachability
h2 -> h3 X X h5
h3 -> h2 h1 h4 X
h1 -> X h3 h4 h5
h4 -> X h3 h1 h5
h5 -> h2 X h1 h4
*** Results: 30% dropped (14/20 received)
mininet> pingall
```

Figure 3. Result of simulation of Identity-based Access from experiment.

From scenario 1, as seen in Figure 3;

1. H1 is blocked from communicating with H2 (level 1 clearance), vice versa
2. H2 is blocked from communicating with H4 and H1 (level 3 clearance)
3. H3 is blocked from reaching H5, vice versa

Figure 3 above shows the dropped packets denoted with "X" marks. H1 and H4 hold shared resources available to say, level 3 clearance holders (H3, H5). H2 cannot access H1 and H4 since H2 holds a level 1 clearance (lowest clearance level). However, H3 and H5 are in different departments, hence are restricted from communicating with each other. H2 is possibly an external vendor, who needs information from H3 and H5, hence, access is exceptionally granted from H2 to H3 and H5.

B. *Scenario 2*

This scenario built on scenario 1 with a little addition to simulate Device Authentication provisioning. Assuming that H5 has not been verified and/or authenticated; hence, H5 was denied communication to any other device in the network.

```
mininet> pingall
*** Ping: testing ping reachability
h2 -> h3 X X X
h3 -> h2 h1 h4 X
h1 -> X h3 h4 X
h4 -> X h3 h1 X
h5 -> X X X X
*** Results: 60% dropped (8/20 received)
mininet>
```

Figure 4. Result of simulation of Device Authentication from experiment.

Once again here, dropped packets are denoted with "X" marks as seen in Figure 4. Notice though the fact that H5 has no successful communication with anyone else (H1 to H4) in the network because though recognised on the network, has not be verified yet. This is the addition that was made to what was done in scenario 1. So, imagine that H5 was being used by a staff with level 3 clearance, in the event that H5 is no longer a valid user or temporarily not in use, it would most likely not pose a threat to the network infrastructure; since, it has to be verified before it can get access to the network devices.

The experiment assures that information is made available to only those who needs it (need-to-know basis), which therefore means that information is readily available as needed (Availability); especially seen in scenario 1 where sufficient access was given to just what is needed. The confidentiality and integrity are illustrated in scenario 2 where H5, though with level 3 clearance access, is prevented from communicating in the test environment – unable to change/ alter/ access any information until it is verified as a legitimate device through Device Authentication provisioning.

V. Conclusion And Future Research

In this paper, credence is given to the integration of SDN and SDP as one holistic solution enabling the design, centralised manageability, and of course security of a network. First, using an extensive review of related works, the stage was set for work done in this research. The definition of the SDN and the SDP concepts was considered; problems and solutions were also considered. The literature review was then narrowed down to papers [1] and [2]. Afterwards, under the Contribution Explained section, some possible security concerns/ vulnerabilities that brought to the limelight the need for a security solution was highlighted. This paper further discussed why SDP was proposed as a security solution for SDNs as opposed to a traditional perimeter security model. In the experiment, Mininet components were built on to design, manage and secure the experimental network. Building on Mininet component, two scenarios were simulated that demonstrated Identity-based Access (SDP feature) and Device Authentication provisioning (SDP feature).

Though this research paper gave attention to all three aspects of the CIA triad, the experiment did not fully address Availability (single point of failure; heavy traffic/ use). A distributed implementation of the proposed system in this paper can be investigated towards mitigating the issue/ concern of single point of failure and availability resulting from link congestion. The experiment in this research was on a SDN (making it provide an extended security function (SDP)); how about having a SDP solution with extended functions of network design, setup and manageability? Or better yet, designing an entirely new system that caters to both SDN and SDP needs – having security built in from inception, as opposed to having security as add-on function?

ACKNOWLEDGMENT

The research was guided by Professors Pavol Zavarsky and Sergey Butakov. Thank you to all the authors whose work I have referenced.

REFERENCES

[1] Ahmed Sallam, Ahmed Refaey, Abdallah Shami, "On the Security of SDN: A Completed Secure and Scalable Framework Using the Software-Defined Perimeter," IEEE 2019.

[2] Ala Darabseh, Mahmoud Al-Ayyoub, Yaser Jararweh, Elhadji Benkhelifa, Mladen Vouk, Andy Rindos, "SDSecurity: A Software Defined Security Experimental Framework" IEEE 2015

[3] Andrew Stibbards, Sunset Learning Institute Instructor "Software-Defined Networking (SDN), What Is It and How Does It Work" [Online]. Available at: https://www.sunsetlearning.com/training-resources/sli-blogs/software-defined-networking-sdn-work/. [Accessed: 30 Oct. 2018].

[4] D. J. Nikoue, S. Butakov, and Y. Malik, "Security Evaluation Methodology for Software Defined Network Solutions," 2017.

[5] J. Garbis and P. Thapliyal, "Software Defined Perimeter for Infrastructure as a Service," Cloud Secur. Alliance, 2016.

[6] Adaranetworks.com. (2018). Horizon Product Series – ADARA. [online] Available at: http://adaranetworks.com/products-horizon.php [Accessed: 23 Oct. 2018].

[7] N. DNA, "Software-Defined Access", Cisco, 2018. [Online]. Available at: https://www.cisco.com/c/en/us/solutions/enterprise-networks/software-defined-access/index.html. [Accessed: 23 Oct. 2018].

[8] P. Services and N. Segmentation, "Cisco Identity Services Engine (ISE)", Cisco, 2018. [Online]. Available at: https://www.cisco.com/c/en/us/products/security/identity-services-engine/index.html. [Accessed: 23- Oct- 2018].

[9] J. Koilpillai, "Software Defined Network (SDN) or Software Defined Perimeter (SDP) … what's the difference?" Waverley Labs LLC., 2016. [Online]. Available at: http://www.waverleylabs.com/software-defined-network-sdn-or-software-defined-perimeter-sdp-whats-the-difference/. [Accessed: 30 Oct. 2018].

[10] L. Miller and P. Gregory, "Common Access Control Models You Should Know for the CISSP Exam", www.dummies.com, 2018. [Online]. Available at: https://www.dummies.com/programming/certification/common-access-control-models-know-cissp-exam/. [Accessed: 26 Nov. 2018].

[11] Cisco "Cisco DNA at a glance" Cisco, 2018. [Online]. Available at: https://www.cisco.com/c/dam/en/us/solutions/collateral/enterprise-networks/cisco-digital-network-architecture/cis-dna-aag.pdf?oid=aagen000309/. [Accessed: 26 Nov. 2018].

[12] NIST Special Publication 800-53 Rev. 4, AC-4, "Information Flow Enforcement", Security and Privacy Controls for Federal Information Systems and Organizations, 2013.

[13] NIST Special Publication 800-53 Rev. 5 (DRAFT), PE-22, "Component Marking", Security and Privacy Controls for Infrmation Systems and organizations, 2017.

[14] Scott Hogg, "SDN Security Attack Vectors and SDN Hardening", CORE NETWORKING, 2014.

[15] Wenjing Lan, Fangmin Li, Xinhua Liu and Yiwen Qiu, "A Dynamic Load Balancing Mechanism for Distributed Controllers in Software-Defined Networking," 10th International Conference on Measuring Technology and Mechatronics Automation, 2018

[16] Ming-Tsung Kao, Bo-Xiang Huang, Shang-Juh Kao, Hsueh-Wen Tseng, "An Effective Routing Mechanism for Link Congestion Avoidance in Software-Defined Networking," International Computer Symposium, 2016

[17] Hui Xu, Jun Su, Xinlu Zong, Lingyu Yan, "Attack Identification for Software-Defined Networking based on Attack Trees and Extension Innovation Methods," The 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2017.

[18] Akash Srikanth, P. Varalakshmi, Vignesh Somasundaram, Pavithran Ravichandiran, "Congestion Control Mechanism in Software Defined Networking by Traffic Rerouting," Proceedings of the Second International Conference on Computing Methodologies and Communication (ICCMC), 2018.

[19] Daria Mostovich, Pavel Fabrikantov, Andrei Vladyko, Mikhail Buinevich, "High-Level Vulnerabilities of Software-Defined Networking in the Context of Telecommunication Network Evolution," IEEE, 2017.

[20] J. Koilpillai, "Input to the Commission on Enhancing National Cybersecurity", NIST The Commission on Enhancing National Cybersecurity, 2016.

[21] J. Koilpillai, "Software Defined Perimeter (SDP) Implementation" Waverley Labs LLC., 2016. [Online]. Available at: http://www.waverleylabs.com/software-defined-network-sdn-or-software-defined-perimeter-sdp-whats-the-difference/. [Accessed: 30 Oct. 2018].

[22] Salaheddine Zerkane, David Espes, Philippe Le Parc, Fréderic Cuppens, "Vulnerability Analysis of Software Defined Networking", 2018.

[23] Don Orlik, "Cisco DNA Center, the evolution from traditional management to intent based automation and assurance", Cisco Connect Edmonton, 2018.

[24] POX Manual [Online]. Available at: https://noxrepo.github.io/pox-doc/html/ [Accessed: 30 Oct. 2019].

[25] A. Mubayed, A. Refaey, A. Shami, "Software-Defined Perimeter (SDP): State of the Art Secure Solution for Modern Networks," IEEE Network Magazine, 2019

[26] P. Kumar et al., "Performance Analysis of SDP for Secure Internal Enterprises," Proc. IEEE Wireless Commun. Networking Conf. (WCNC'19), Apr. 2019