مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (40) العدد (6) 2018
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (40) No. (6) 2018

Implementing Micro-Fragmented LAN Network Security Plan using Network Virtualization (NV) and Software Defined Network (SDN)

Dr. Sira ASTOUR[1]

(Received 10 / 4 / 2018. Accepted 11 / 11 / 2018)

□ ABSTRACT □

Legacy computer networks' security and access rules rely on previously defined segments in the physical networks which usually leads to many, coarse-grained, hard to change security rules. New technologies of network virtualization, programmable switches and Software Defined Network (SDN) allow the use of better approaches for securing networks. This is especially crucial for the network portions that are not segmented, such as LANs, or inside one segment in a data center. We implemented an inside segment, cross cutting security rules on a proposed network using the new previously mentioned technologies. The implemented security rules are designed to be fine-grained, classless, and segment free that could work on multiple levels of the network reference model, or on the host port level inside a LAN at the same time. This was done in order to explore and show the benefits of using Network Virtualization (NV) and (SDN) technologies to achieve micro-fragmented security plans. A security plan scenario was designed in a way that demonstrates multiple network layers security objectives, and cross cutting access rules to multiple network segments. These segments were defined physically, and by using virtual networks' tags (VLAN).

The suggested network were implemented using the Mininet simulation for SDN, and the POX controller after adding the suitable code to realize the suggested security plan. Results show the success of implementation of fine-grained, segments cross-cutting security rules, the ease and flexibility of applying such rules on-line, the dynamicity of it, and its adaptability with any changes applied to the proposed network.

Keywords: Access Control, Micro-fragmentation, Network Virtualization (NV), OpenFlow (OF) Protocol, Mininet, and Software Defined Network (SDN).

[1] **Teacher - Faculty of Communication and Information Engineering – Arab International University – AIU, Damascus**
Charger of Affaires – Faculty of Informatics Engineering – Damascus University

مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (40) العدد (6) 2018
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (40) No. (6) 2018

تحقيق خطة حماية شبكة محلية ذات تجزئة ميكروية باستخدام مفاهيم افتراضية الشبكة والشبكات المعرّفة برمجياً

الدكتورة سيرا أستور[2]

(تاريخ الإيداع 10 / 4 / 2018. قُبِل للنشر في 11/ 11 / 2018)

□ ملخّص □

تعتمد حماية الشبكات الحاسوبية التقليدية وقواعد النفاذ المستخدمة فيها حالياً على تعريف مسبق لمقاطع ضمن الشبكة الحاسوبية المستهدفة؛والذي عادة ما يقود إلى عدد كبير من قواعد النفاذ ذات الحبحبة الخشنة (Coarse-Grained) والتي تكون غير مرنة في التعامل، ومرتفعة الكلفة لتطبيق التغييرات والتعديلات على الشبكة خلال تطورها وتغييرها مع الزمن. تسمح التقانات الحديثة في الشبكات الحاسوبية مثل تحقيق الافتراضية في الشبكات (NV)، والشبكات المعرّفة برمجياً (SDN) باستخدام مقاربات أفضل لتحقيق قواعد حماية الشبكة وبالأخص على الشبكات المحلية؛ وتمكّن من تحقيق الحماية على مستوى عناصر الشبكة وعلى أكثر من طبقة من طبقات الشبكة المرجعية بدلاً من حدود مقاطعها المعرّفة مسبقاً إما فيزيائياً، أو حتى إن كانت معرّفة عن طريق الشبكة الافتراضية VLAN. تم في هذا البحث تطوير خطة حماية ذات تجزئة ميكروية تخترق عدة مقاطع شبكية معرّفة مسبقاً.هذه المقاربة الجديدة في الحماية لا ترتبط بطبقات الشبكة التقليدية مثل طبقة الشبكة أو طبقة ربط البيانات أو التطبيقات، وتوفّر قواعد حماية مختلفة على مستوى مضيف و/أو بوابة محددة من مضيف. تم تحقيق خطة الحماية على شبكة بسيطة تم تعريفها باستخدام برنامج المحاكاة (Mininet)، الأكثر شهرة عند التعامل مع تقانات الشبكة المعرّفة افتراضياً (SDN)، وتم التحكم بالدفق وقواعد النفاذ عبر استخدام المتحكم البرمجي (POX)، وتم إضافة الترميزات المناسبة للتحكم بقواعد النفاذ المختلفة. وأظهرت النتائج نجاح عمليات الحماية الميكروية الجزئية والمقدرة العالية لتحديث وتعديل الخطط الأمنية بفاعلية وأداء عال.

الكلمات المفتاحية: التحكم بالنفاذ، التجزئة الميكروية، تحقيق افتراضية الشبكية، بروتوكول (OpenFlow)، الشبكات المعرّفة برمجياً (SDN).

2 دكتورة مدرسة في كلية المعلوماتية والاتصالات – الجامعة العربية الدولية –دمشق
قائم بالأعمال في كلية الهندسة المعلوماتية – قسم النظم والشبكات الحاسوبية – جامعة دمشق

I. Introduction

Traditional approaches for securing computer networks are not suitable anymore at our present time, due to the vast spreadof clouds, service on demands, and Data Centers (DCs). Moreover, they are ineffective for handling today's ever strengthening security threats such as Advanced Persistent Threats (APTs), and coordinated attackswhich often include months of monitoring, reconnaissance, vulnerability exploits, and hidden inactive malware agents that could activated remotely [7].

The new security approaches of networks should assume that threats can be anywhere and everywhere; which acquires redefining the security model of networking along with rethinking networking paradigm implemented and used in recent years. This is augmented with the adventof programmable networks and Software Defined Networks (SDNs).

As a part of that, traditional perimeter firewalls implemented today assume inside trust and require all traffic to pass through it;causing choking points in large and heavily traffic networks. Virtual LAN (VLAN) is used to apply segmentations that could offer another inside level of security rules; however, any penetration of the network using a backdoor or unauthorized access using a legal port, the threat could easily and freely move from one machine to another.

Sensitive data could not afford such vulnerability; especially with the vast use of Data Centers that might hosts different workloads for different applications belonging to different companies with different security requirements and demands on the same machines and on the underlying networks.

Usually distributed firewalls are used in traditional networks to offer in-LAN security by using many, expensive, not easy to configure, and vender-specificdifferent machines and software. This is done to enable additional security checkpoints inside LANs for traffic passing from on node to another. This solution is highly expensive, complex, usually require box-to-box configuration from the network administration, and highly inflexible for changes in network topology, or configuration, or any changes in the security policy implemented.

A micro-granular security plan is needed to tie security to micro-fragmentations of networks, and to individual workloads when the network is a service provider to offer agility to provision policies automatically.

Network Virtualization (NV) has the potentiality to enable micro-granular security. Since it enables replicating all the network physical nodes and devices in software; such as logical switches, logical routers,creating an overlay for the physical networks.It isolates, by default, the created virtual networks, and different workloads of the networks from each other, and from the underlying physical infrastructure.

Since, isolation is a key factor in security, NV needs no additional physical subnets, or VLAN, or Access Control Lists (ACLs), or firewall rules to ensure isolation between different networks or/and different workloads; or between networks and the physical infrastructure beneath.

In this paper we are exploring the benefits of using the mentioned techniques to implement dynamic,and flexible security policies for a private LAN network using a micro-fragmented security plan. The aim is to experiment through implementation and results the agility, flexibility solutions which are inexpensivethat could be obtained of such scenario.

The importance of the research and its objectives

The importance of research derived from the possibility of using the most important modern technologies to provide the protection of local networks. The problem of LAN protection still representing vulnerable and problematic issues in the current networks. The first aim of the research is to achieve a local network protection plan designed to protect a number of network layers at the same time, and apply multiple access rules that by pass different network segments. The second goal is to demonstrate the possibility of making updates to access rules easily and flexibly through the programmatic control of the network realized and implemented virtually.

Research methods and materials

The research realize the proposed network by using a modern network simulation tool for SDN called Mininet on Linux server. In addition, we used a standard programmable network controller which is (POX), that has been programmed to achieve the security proposed scenarios and providing the micro-fragmented cross cutting protection levels. We investigated and implemented several scenarios to derive the results presentedthat fulfill the objectives of this paper.

II. Problem of legacy networks

Traditional networks uses old protocols,especially in DCs, like Spanning Tree Protocol (STP) to prevent routing loops, and broadcast radiation by disabling some ports in the networks that are part of that tree. Disabled ports means that a large parts of the networks could be in a passive mode which leads to highly inefficient, underutilized networks.

Moreover, legacy networks are static; this makes network infrastructure provisioning to take a long time, and they could not cope with the dynamic nature of today's business demands. On the other hands, static nature of the networks makes implementing changesin the networks, including the security changes, to be done on a manual, box-to-box basis which is highly human error prone. Nowadays, mis-configuration and applying changes on traditional networks cost a long downtime of networks operations.

Other problems in traditional networks include that they are designed for best-effort traffic; such as e-mail and internet browsing. New technologies are been used todayto permit a guaranteed Quality of Service (QoS) to users; but those technologies are not applicable to old networks. Moreover, connecting distributed parts of a VLAN over the Internet, which is a third layer network and works on IPs.This creates the problem of a packet losing all of its layer-two headers when it crosses the internet, including the VLAN tags. This limitation of VLAN means that we lose the isolation of a private network when the traffic crosses the internet that could expose the private networks tomany different threats and problems, not mentioning the limitations of VLAN numbers (e.g. 4096) andits static allocation nature.

Developing solutions to these problems needs lots of protocol overlaying; where some protocols and additional features need to be implemented at layer-2 and/or on layer-3. This usually leads to increasing the complexity of the network management and affecting the overall performance of it.

With the vast deployment of server virtualization, clouds, services on demands, and real time collaboration, the traditional networks slow evolving process presents an obstacle to leveraging new services and fulfilling demands of the business worlds.

III. Network Virtualization (NV)

Virtualization technology unlike the server-client model, allows multi-tenant environments. These environmentsprovide computing resources (such as CPU, memory and storage) to be shared by different applications.This would increase resource utilization and efficiency and lower the Operational Expenditure (OPEX) of networks.

Virtualization by definition is about creating an abstract of the underlying system components away from the hardware details and presents logical view of these resources. The goals of virtualization includes achieving a higher level of performance scalability, andreliability, and agility or to create a unified security and management domains.
Kysnetzky [1] provided a model of virtualization stack includes seven layers of different virtualization types as shown in figure-1. In addition, NV creates an overlay of logical networks over the physical resources rather than physically connecting two domains in a network. It replicates all the physical nodes and devices in software. It works by creating a tunnel between the virtual logical networks and the physical ones separating them from each other and using the underlying physical infrastructure as a simple packet forwarding backplane. NV is valuable because it saves administrators from having to physically wire up each new domain connection; especially for virtual machines that get created in a virtualized environments.

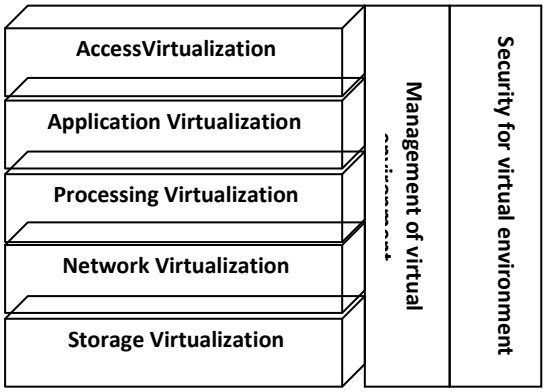


Fig. 1. Virtualization Stack Model [1]

In NV,Virtual Machines (VMs) are connected logically to each other and to the VN to transfer data through virtual switches. A (vSwitch) works a much like a physical Ethernet switch connecting VMs on different virtual ports.It detects which VM is connected to which port to forward data to the corrected VMs(figure-2). A vSwitch can be connected to physical Ethernet adapters (uplink adapters) to connect Virtual Network (VN) to physical one. Usually, they are embedded into installed software, or could be included in a server hardware as part of its firmware. Almost all new versions of operating systems are already including virtual switches on them or on a hypervisor; including Linux, Microsoft and VMware among others [2].
There are many new overlaying encapsulation technologies used to provide virtualization of the networks. One industry standard technology is the Virtual Extensible Local Area Network [3], or (VXLAN)which provides a framework for overlaying virtualized layer-2 networks over layer-3 networks. The original specification of this technology is created by VMware [4], Arista Networks [5], and Cisco [6]. Many of the leading companies of networking are backing this standard.

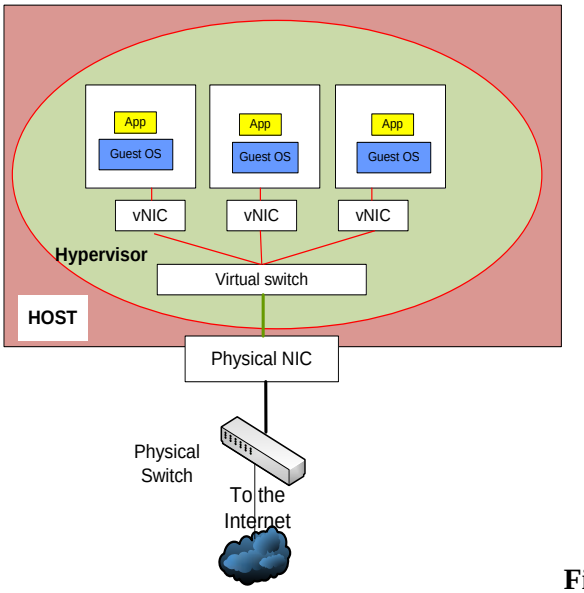


Fig. 2.The Virtual Switch

Another known overlay technology is the NVGRE- Network Virtualization using Generic Routing Encapsulation (GRE), which is similar to VXLAN in its purposes, but it uses different approaches to create the overlay. NVGRE used in Microsoft has had limited adoption in comparison to the momentum of VXLAN that is used in VMware and other companies.

Hypervisors nowadays implement virtual switches (vSwitch) to enable NV. Each VM has at least one virtual network interface cards (vINCs) that are sharing physical Network Interface cards (pNICs) on the physical host through vSwitcs (figure-2).

The original vSwich had many deficiencies; among themnot supporting VLAN, port mirroring, and port channeling. This prevented network administrators from separating packets from different VM users; so all VMs residing on the same physical machine have their traffic visible to each other, which imposed a high security threats.The new versions overcame these problems.

IV. Software Defined Network (SDN)

Transport network protocols and distributed control running inside the router and switches are the mean components enabling the passing of the network traffic from one node to another, and from one network to another, over the internet. Even though IP protocol and networks are widely adopted and used; they are hard to manage, inflexible to changes and poses many limitations of network evolvement.

Implementing high-level network policies, network administrators and operators have to do box-by-box configurations on every device on the network using low-level and often vender-related specific commands.

Software Defined Networking (SDN) is a networking paradigm that separates the control plane from the data (forwarding) plane, centralizes the network control, and defines open, programmable interfaces [7].

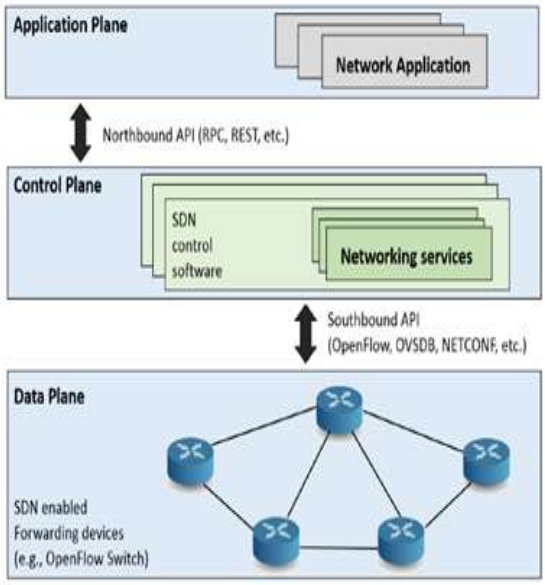


Fig.3. SDN architecture Overview

A. OpenFlow Protocol

OpenFlow (OF) provides the standards for the interface on controlling the data packets. Open Network Foundation (ONF) has goals of setting global standards and interoperability in place. "The OpenFlow standard also provides a basic set of global management abstractions, which can be used to control features such as topology changes and packet filtering [8]."

OpenFlow enabled-switch could be broke down into three main parts; the flow table, secure channel, and the OpenFlow protocol. It uses the concept of flows to identify network traffic based on pre-defined match rules that can be statically or dynamically programmed by the SDN control software. It also allows ITs to define how traffic should flow through network devices based on parameters such as usage patterns, applications, and cloud resources.

Since OpenFlow allows the network to be programmed on a per-flow basis, an OpenFlow-based SDN architecture provides extremely granular control, enabling the network to respond to real-time changes at the application, user, and session levels. Current IP- based routing does not provide this level of control, as all flowsbetween two endpoints must follow the same path through the network, regardless of their different requirements. [9]

The open standard OF protocol abstracts the underlying physical network which forwards the payload data. It matches on arbitrary bits in packet, and execute actions such "Forward to port", "Drop", "Send to controller", and "Mangle packet" among others. Moreover, OF enables networks to evolve, by giving a remote centralized controller the capability to alter the behavior of network devices, through a well-defined "forwarding instruction set". The growing OpenFlow ecosystem now includes routers, switches, virtual switches, and access points from a range of vendors.

B. SDN Controllers

The control plane takes the distributed control of the Internet by centralizing the control of an entire physical SDN network in a single logical centralized SDN controller (or a network Operating System). The main task of the SDN controller is to set the routing rules

to be followed by each forwarding device and to decidewhat is the best for the network based on a global view of the network.
The controller does that through standardized interfaces, called the south-bound interfaces. These interfaces can be implemented using protocols such as OpenFlow 1.0 and 1.3, or OVSDB [10] and NETCONF[11]. The control plane concentrates, thus, the intelligence of the network, using information provided by the forwarding elements (e.g., traffic statistics and packet headers) to decide which actions should be taken by them [12].
Different types of SDN controllers exist in the ecosystem. The first SDN controller was NOX, which was initially developed by Nicira Networks, alongside OpenFlow. In 2008, Nicira Networks (acquired by VMWare) donated the open source NOX to the SDN community, where it has become the basis for many subsequent SDN Controller solutions [13].
NOX is a C++ multi-threaded controller that is written on top of a POX library, which is a single threaded python controller that will be used in our implementation. Beacon is another Java basedcontroller [13], and many more variations.
Each type of controller has its strengths andweaknesses; however, the choice of the controller depends on the needs, languages, among other factors.

V. Network Virtualization using SDN

Even though network virtualization and SDN are independent concepts, the relationship between these two technologies has become much closer in recent years.
Using the OF protocol, a hypervisor can now establish multiplevirtual SDN networks (vSDNs) based on a given physicalnetwork. Each vSDN corresponds to a "slice" of the overallnetwork. The virtualization of a given physical SDN networkinfrastructure through a hypervisor allows multiple tenants (such as service providers and other organizations) to share theSDN network infrastructure. Each tenant can operate its ownvirtual SDN network, i.e., its own network operating system,independent of the other tenants.
While NV adds virtual tunnels and functions to the physical network, SDN changes the physical network. This is therefore really a new externally driven means to provision and manage the network. SDN is implemented on network switches, rather than x86 servers. BigSwitch[14], and Pica8[15] are examples of companies selling SDN-related products.
NV leveraged Software-Defined Networks (SDNs) making the management tasks easier and programmable. Software Defined Security (SDS) operates networks as single enforcement domains with every elements became enforcement points. Many controllersnow are already addressing some of the security general issues such as simple authenticated communication channels and control data replication among different controllers 'instants.
In addition, SDN technology offers both new opportunities and challenges. SDN advantages includetheenhancementof network security when the control plane act as packet monitoring and analysis instrument that is able to propagate security policies such as access control [17], along the entire network in response to attacks [16]. In addition, with the higher control over packets routing which is provided in SDN, we can install security appliances such as firewalls and IDS in any part of the network, not only on its edges [18]. This is possiblesince controllers steer the corresponding traffic to designated nodes, the packets can be analyzedand treated accordingly.
This flexibility is, for example, at the core of the Software Defined Perimeter (SDP) concept [19], which applies authentication and authorization rules on every entities and devices trying to access a given network infrastructure. Itis also crucial to thwart denial-of-service (DoS) attacks, since then the task of discarding malicious packets is not

concentrated on one or a few security devices near the attack's target, but distributed along the network [20].

In additionSDN could create the illusion of a "moving target" by continuously changing the host persistent address to different IPs over time; so it could defend the network from a DoS or other threats targeting the host static IP [21].

A. Open vSwitch

Open vSwitch (OVS) is a software-based solution that resolves the problems of network separation and traffic visibility, so users can be assigned VMs with elastic and secure network configurations.Moreover, it enables flexible controller in user-space and fast datapath in kernel.OVS [22] is an example of a software-based virtual network switch that supports VXLAN overlay networks.

B. Basic features of OVS

In OVS [22], packets are managed as flows; a flow may be identified by any combination of input port, or VLAN ID (802.1Q), or Ethernet Source MAC address, Ethernet Destination MAC address, IP Source MAC address, IP Destination MAC address, TCP/UDP/... Source Port, and TCP/UDP/... Destination Port.

OVS has multiple ports to physical switches where a port may have one or more interfaces, Packets are forwarded by flow, and it supports IEEE 802.1Q which means that it enables virtual LAN function by attaching VLAN ID to virtual interfaces, so each user will have its own LAN environment separated from other users (figure-4). This isolation function enable fine-grained ACLs and QoS policies through extensive flow matching capabilities and possible chains of actions

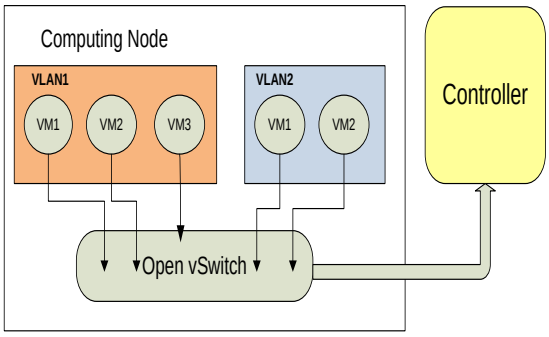


Fig.4: Open vSwitch Isolation Feature

Flow matching includes identifiers from layer-1 to layer-4 of networks; such as tunnel ID in layer-1, or Mac Address or VLAN ID in layer-2, IPv4/IPv6 fields in layer-3, and TCP/UDP and ICMP in layer-4. The existing of a pipelined chains of actions allows different actions to be applied to different flows such as packet angling.

Two kinds of OVS are in the markets today; software switches such as Open vSwitch (OVS), Cisco Nexus 1000V, and hardware switches such as Brocade, Cisco, HP, IBM, Juniper Networks, NEC, and others.Thereis a third portion for users; it is switching ASICs; such as Indigo which is anopen source firmware leveraging Ethernet switch ASICs to support up to 48x 10G ports, and Mellanox SwitchX-2 chip [22].

VI. Design and Implementation

The test bed consists of different software pieces; a virtualization software (VMware workstation 9.0) which includes an OVS implementation, and the POX controller for the control plane. POX is an open source development platform for Python-based

softwaredefined networking (SDN) control applications, such as OpenFlow SDN controllers. It enables rapid development and prototyping.
Another piece of used software is the Mininet; which is an emulator for deploying large networks on the limited resources of a simple single computer or Virtual Machine (VM). It allows running unmodified code interactively on virtual hardware on a simple PC. Also, it provides convenience and realism at very low cost.
The alternative to Mininet is hardware test beds which are fast, accurate but very expensive and shared. The other option is to use simulator which is very cheap but sometimes slow and require code modification.
The used approach is by implementing the security rules in the controller itself. In this way, an authenticated network administrator would have an access on the control plane (e.g., the controller or the controllers' front end) and be able to apply, monitor and to change the security rules by experimenting on real time the established network. We are doing this without an application level appliance for such a sensitive service. This model makes the local network harder for outsiders and unauthorized entities to expose the vulnerabilities within.

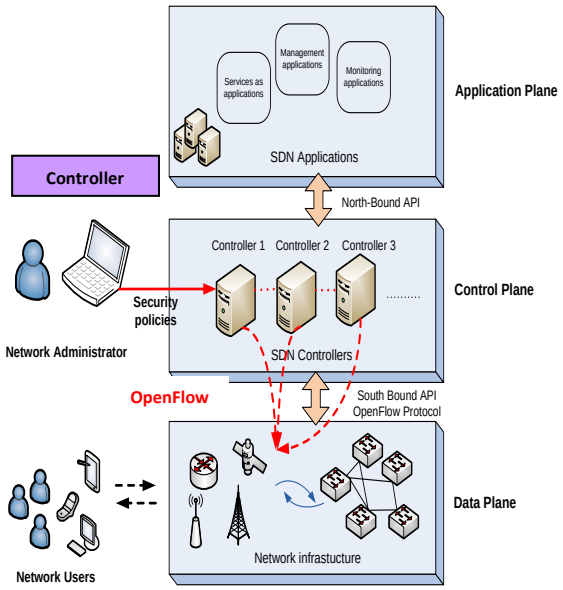


Fig. 5. Basic proposed network architectures

The applied security policy is a generic document that outlines rules for computer network access.It determines how policies are enforced and lays out some of the basic architecture of the company security or network security environment. We implemented our suggested security policy by writing it to a (.csv) file which is read by the controller at the start up or at any security policy change.
The network we proposed is for a company with differentadministrative departments, each has different intersectional security requirements per host, or per groups of hosts, and not per department. Different security rules are in need on multiple layers. In this experiment we propose rules on layer two of the network stack by blocking MAC addresses and a specific port (the internet port), and on layer three, by blockingIP addresses.
We also defined a VLAN with hosts from different departments. The design security plan consisted of multiple parts that included different layers of the network protocol stack such as layer-2 and layer-3.

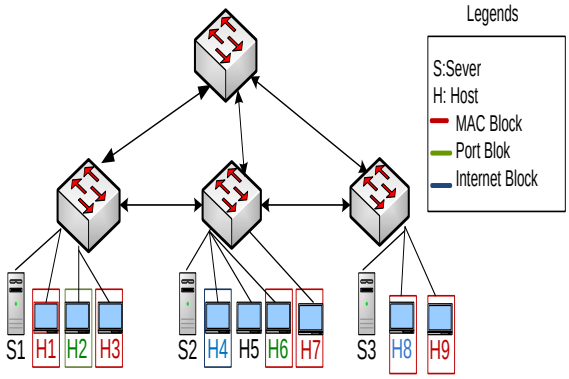


Fig. 6. Thesuggested security plan for the proposed network

The rules included intersecting blocking on MAC level and on IP level among different hosts. In addition the security implemented plan included preventing different hosts form accessing the Internet while allowing the others (Fig-6).

The security plan designed of fine-granular security rules that crosses different departmental domains and the defined VLAN to demonstrate the aim of this experiment. The security plan was written into a CVS file and read by the controller on startup of the network.It also could be pushed into it when the security rulesare changed or updated.

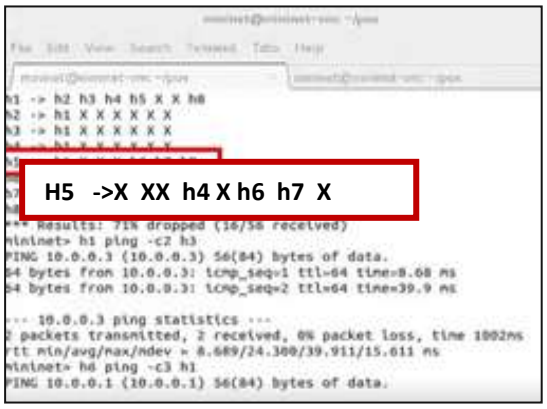
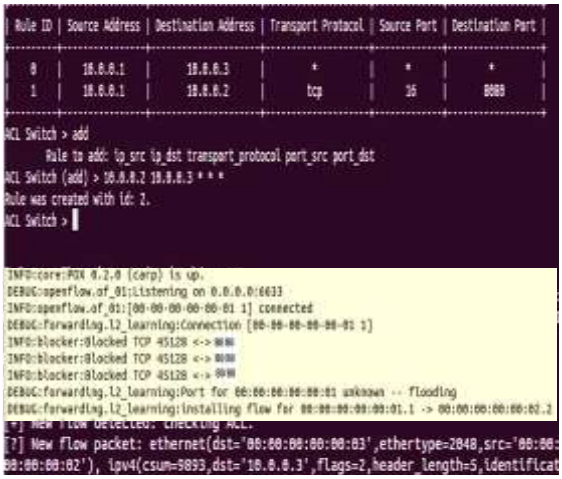


Fig. 7. Applying the security policy on MAC and IP addresses



After succeeding in this experiment we changed the security rules creating another protection scenario. The new rules were written with the intention to apply a major changes to the old already implemented ones.
The new experiment was done to highlight the ease of updating or changing the security plan based on the change of the physical network, or the change of ACLs for different employees in the company, or any other administrational changes. The new plan was rewritten into the CVS file and easily implemented on the click of a mouse. Again the controller sent all updated security rules to enabled switches and new rules are set and implemented for the networks.

VII. Related Works

A few related works exist on implementing security rules using the logical central programmable networks such as 4D[24], SANE [25], Ethane[26], and Resonance [17]. In 4D[24] a clean-slate approach is proposed called the "4D approach," named after the four planes of decision, dissemination, discovery, and data. The 4D architecture completely re-factors the functionalities of a network and separate the network control from the forwarding substrate. The authors propose that a network architecture should be based on three key principles i.e., i) network-level objectives, ii) network-wide views, and iii) direct control [24].
SANE [25] is a Secure Architecture for the Networked Enterprise. It is a clean-slate protection architecture for enterprise networks. The aim of SANE includes an architecture thatsupports simple but powerful natural policies, independencefrom topology and network equipment, link layer security, protectionof topology and services information from unauthorizedaccess, and centralized definition and execution of all the policies [25]. The SANE architecture has a Domain Controller(DC) that performs authentication of network entities, advertising available services and controlling connectivity in the network.
SANE work has been extended to Ethane [26].Ethane also works on automating security rules using centralized programmable system, but it focuses on host authorization as opposite to our work which focuses on micro-segmented security policy.
Resonance [17] is the closest to our system; it also works on access control and dynamic policy enforcements but it focuses on continuous monitoring and interference-based policy.
In the paper with the title "Security in Software Defined Networks: A Survey" [27],most of the mentioned works are categorized and compared together.

VIII. Conclusion

Security rules could be formulated and implemented on a micro-fragmented manner which allows a substantial improvements over traditional security models for a single segmented networks, or the network inside a datacenter. Further additional rules related to all network protocol stack levels could be added such as application level rules to demonstrate the efficiency of such new approaches.

REFERENCES

[1] KUSNETZKY D. K.,*Virtualization is more than Virtual Machine Software*. Available at *http://www.kusnetzky.net/publications/ImpactPapers/20070829_Virtualization_is_more_than_VM.pdf* - 2007.
(Last visited: November 2017).
[2] EMMERICH P., RAUMER D., WOHLFART F., and CARLE G.,*Performance characteristics of virtual switching.* In Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on. IEEE, 2014.
[3] MAHALINGAM M., DUTT D., DUDA K., AGARWAL L.,KREEGER P., SRIDHAR T., *A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks.*RFC7348: Virtual eXtensible Local Area Network (VXLAN), 2014.
VMware Documentation; *https://www.vmware.com/support/pubs/*.(Last visited: November 2017).
Arista - Software Driven Cloud Networking*; https://www.arista.com/*.(Last visited: February 2018).
[4] Publications | Cisco Research Center; research.cisco.com/publications.
[5] RAGHAVAN B.et al., *Software-defined internet architecture: decoupling architecture from infrastructure*, in Proceedings of the 11th ACM Workshop on HotTopics in Networking. 2012, pp. 43–48.
[6] KREUTZ D., RAMOS F. M., VERISSIMO P., et al.;*Software-defined networking: A comprehensive survey*, Proc. IEEE, vol. 103, no. 1, 2015, pp. 14–76.
[7] OpenFlow*; http://www.openflow.org*
[8] PFEFF B., DAVIE B., *open vSwitch database management protocol*, IETF RFC, 7047, 2015.
[9] ENNS R., BJOVKLUND M, et al., *Network Configuration Protocol (NETCONF),*Network Configuration Working Group's proposal of NETCONF, RFC 624, IETF, June 2011.
[10] KREUTZ D., RAMOS F., and VERISSIMO P., *Towards secure and dependable software-defined networks*, in Proceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, Hot SDN 2013, New York, USA, ACM, August 2013, pp. 55–60.
[11] What are SDN Controllers?, SDX Central," [online] 2014,
https://www.sdxcentral.com/resources/sdn/sdn-controllers/(Last visited: November 2017).
[12] Big Switch Networks, Inc.*www.bigswitch.com/*(Last visited: November 2017).
[13] Pica8*www.pica8.com/*(Last visited: November2017).
[14] SEZER S. et al., *SDN security: A*
Survey, in Proceedings of IEEE for Future Networks and Services (SDN4FNS), 2013, pp. 1–7.
[15] NAYAK A., REIMERS A., FEAMSTER N., and CLARCK R., *Resonance: Dynamic access control for enterprise networks*. In Proceedings of the 1st ACM Workshop on Research on Enterprise Networking (WREN'09), New York, NY, USA, 2009, pp. 11–18.
[16] YuHunagbC., MinChiT.,YaoTing C., YuChieh C., and YanRen C.,*A novel design for future on-demand service and security*. In 12th IEEE Int. Conf. on Communication Technology (ICCT), 2010, pp. 385–388

[17] BILGER B., BOEHME A., et al.,*Software Defined Perimeter*. Cloud Security Alliance – CSA. 2013.
https: //cloudsecurityalliance.org/research/sdp/. (Last visited: October 2107).
[18] YuHunag C., MinChi T., YaoTing C., YuChieh C., and YanRen C., *A novel design for future on-demand* service and *security*, In 12th IEEE International Conference on Communication Technology (ICCT), 2010, pp.385–388.
[19] JAFARIAN J. H., AL-SHAER E., and DUAN Q., *Openflow random host mutation: Transparent moving target defense using software defined networking.* In Hot Topics in Software Defined Networks –HotSDN. ACM, 2012
[20] Open vSwitch (2015). Production Quality, Multilayer Open Virtual Switch. Last visitedFebruary 2018).
[21] NOXRepo.org (2015). About NOX.
http://www.noxrepo.org/pox/about-nox/. (Last visited:February 2018).
[22] GREENBERG A., MALTZ D. A., et al.,*A clean slate 4d approach to network control and management*. ACM SIGCOMM Computer Communication Review, vol. 35, no. 5, Oct. 2005. pp. 41–54.
[23] CASADO M., GARFINKEL T., et al., *SANE: A Protection Architecture for Enterprise Networks*. In Proceedings ofUsenix Security,2006, pp. 137–151.
[24] CASADO M., FREEDMAN M., et al., *Ethane: Taking control of the enterprise*. In Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'07), Rev., vol. 37, no. 4, NY, USA, 2007,pp. 1–12.
[25] AHMAD I., NAMAL S., et al., *Security in software defined networks: A survey*,IEEE Communication Surveys and Tutorials, 2015.