KADIR HAS UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

Protecting OpenFlow Switches against Denial of Service Attacks

GRADUATE THESIS

Ebada Mohamed Essam Eldin Ibrahim ElDessouky

June, 2017

Protecting OpenFlow Switches against Denial of Service Attacks

Ebada Mohamed Essam El din Ibrahim ElDessouky

Submitted to the Graduate School of Science and Engineering

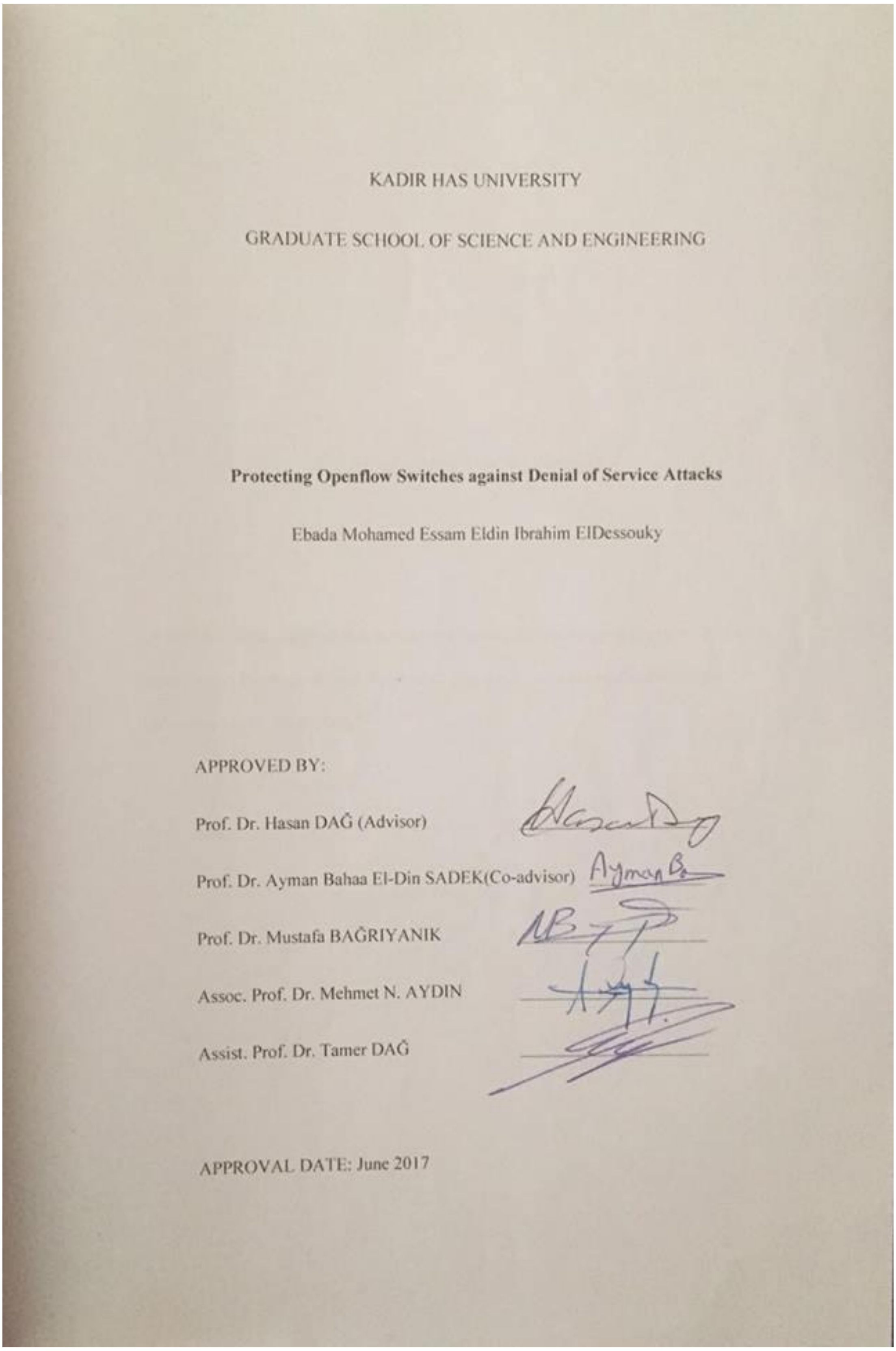
In partial fulfilment of the requirements for the degree of

Master of Science in

Information Technology

KADIR HAS UNIVERSITY

June, 2017



KADIR HAS UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

Protecting Openflow Switches against Denial of Service Attacks

Ebada Mohamed Essam Eldin Ibrahim ElDessouky

APPROVED BY:

Prof. Dr. Hasan DAĞ (Advisor)

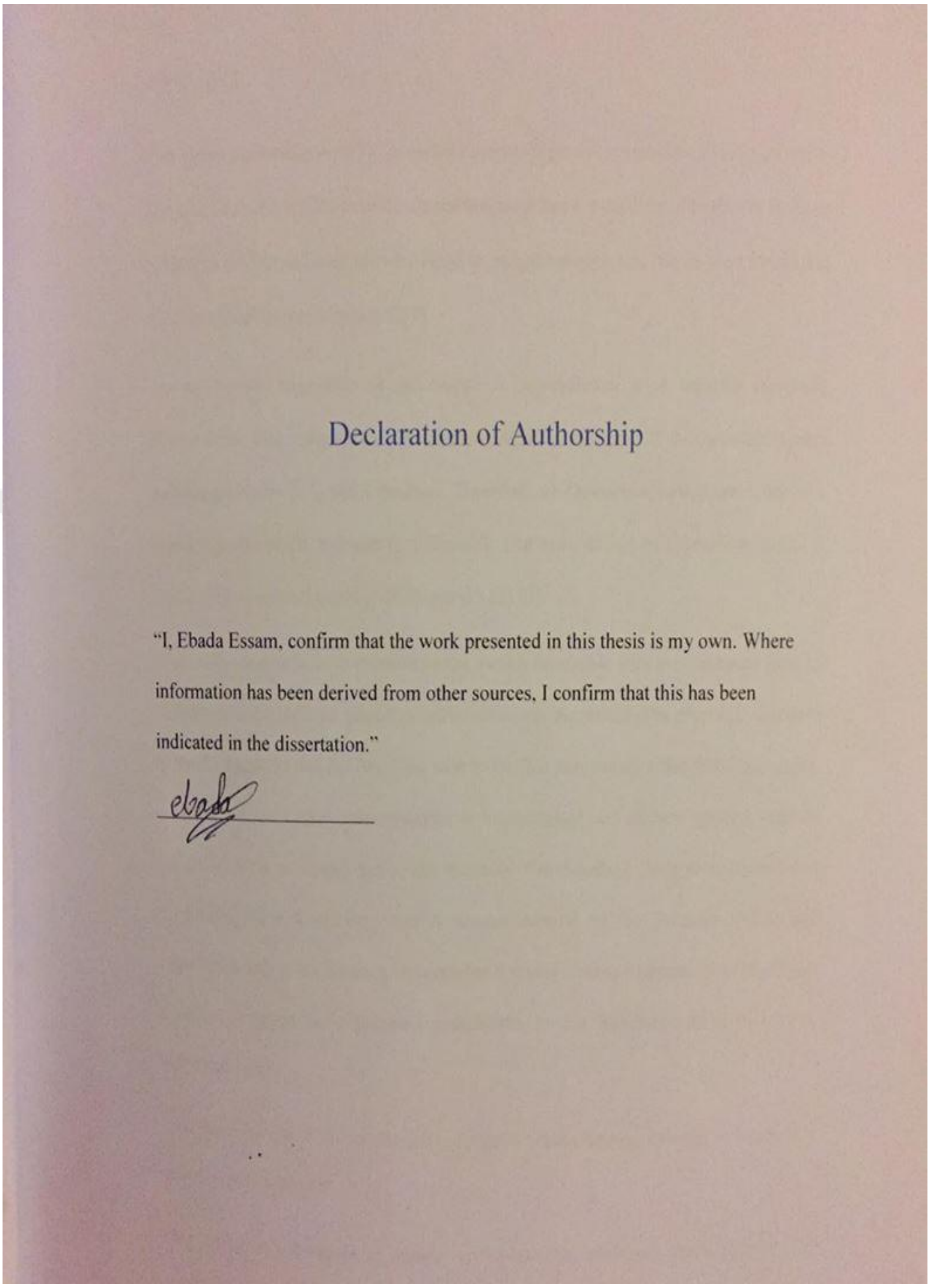
Prof. Dr. Ayman Bahaa El-Din SADEK(Co-advisor)

Prof. Dr. Mustafa BAĞRIYANIK

Assoc. Prof. Dr. Mehmet N. AYDIN

Assist. Prof. Dr. Tamer DAĞ

APPROVAL DATE: June 2017



Declaration of Authorship

"I, Ebada Essam, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the dissertation."

ebada

Abstract

This thesis presents a novel approach to protect OpenFlow switches against a type of Denial of Service (DOS) attacks. OpenFlow switches are the core of Software Defined Networks (SDN) and they are very flexible, programmable, and can be used for several functionalities within a network [1].

As the control algorithm of the switch is implemented on a separate computer (Controller), this software can be implemented on any part of the network packet including Layers 2, 3, and 4 headers. Therefore, an OpenFlow switch can work as a conventional switch, a router or a firewall. The open design of OpenFlow makes it vulnerable to several types of DOS attacks [2] [3].

One of those attacks is to overwhelm the switch flow table with entities larger than its buffer making legitimate packets unable to traverse the switch. The proposed approach is the first time to test the Sandbox with SDN. We propose on a Sandbox like model, where a second switch and controller is implemented and all new packets with no matching rules are forwarded to the Sandbox. The Sandbox clone is monitored and controlled, so a forwarding rule is always created on the Sandbox switch and transferred only to the working switch when it is classified as a normal rule. Otherwise, a clean-up operation is executed periodically on the Sandbox switch to remove malicious rules.

The rules are classified based on the statistics entries already existing in OpenFlow switches flow table [4].

The proposed approach is simple and does not need any extra memory or modifications in the switches. It is proven to mitigate this type of DOS attacks [5].

Acknowledgments

First of all, I praise Allah for all his great and continuous guidance, strength, generosity, and help.

I would like to give my deepest thanks to the very knowledgeable Prof. Dr. Ayman Bahaa El-Din Sadek for his patient teachings, guidance and enthusiastic support in every phase of this dissertation. Also, he helped me to attend seminars to see other graduate students' ideas. Gratefully, his rich knowledge enlightened my path and benefited me greatly.

I would like also, extend thanks Prof. Dr. Hasan Dağ for encouraging me to choose a unique topic and believed that I could do something different. His affectionate welcoming and his sympathy during my master thesis at Kadir Has University was a lifetime experience. And he kindly helped me to attend seminars related to this thesis and integrated me within a group of my colleagues as a beginning to make knowledge and to listen to different topics as having an audience to understand the word of sharing.

My sincere thanks to my family support, encouragement, and care throughout my life is always unforgettable. Learning is an endless path and my professors' guidance is highly appreciated to move up from one point to another.

I have learned more than I can imagine in this journey as it was my first time to be completely dependable on myself and far from my home in the whole period as through brainstorming, laughter's and hard times. It was so difficult to hold on and taking care of everything around me, may I say I finally grow up. Allah endowed a lot of gifts on my way. May this give me the chance to go back home with a head up as I can be useful to the world to be with the help of others.

Table of Contents

Table of Figures

Introduction

Although, networks linger behind the use of old layered innovations with vendor specific interfaces. In this method, it is disorganized and inaccuracies inclined to deal with the networks. In view of the possibility of a unified administration, Software Defined Networking (SDN) addresses this issue by decoupling the information plane and the control plane [1].

The guideline of SDN is basic: a control layer will comprehend current system prerequisites as indicated by the system executive utilizing later and proficient innovations and will broadcast required and classified principles through the information plane layer which can be, for instance, basic switches with a standard interface like the OpenFlow convention [4]. This convention permits the controller layer to indicate rules about how the switch will need to deal with packets from OSI Layers 2 to 4: Ethernet, IP, TCP/UDP, etc. Consequently, the control plane will have the capacity to utilize abnormal state assets like databases, take a shot at standard servers or virtual machines to be effectively modifiable, guaranteeing adaptability and versatility while exhibitions are ensured by the utilization of committed equipment for the switches [6].

Virtual Machines (VM) is a virtual hardware of true implementation that runs on the same software. It’s a chunk of node seen likewise an autonomous machine for its identity or kernel and advantages. Every VM runs a working framework which may be known as guest OS demand an operating system from those groups. VMs run on user-space along these lines around highest priority on the facilitating working framework without hosts, in spite of being mindful that they need support not running for true equipment. As an outcome, SDN address current desires by giving an adaptable furthermore, programmable system which can get data from switches, clients, overseers and furthermore security gadgets and utilize them with a specific end goal to refresh the current strategy as per characterized rules which can now utilize modern capacities. As of now, simply the proposed SDN and OpenFlow begin a far reaching, not just in research works additionally in industry for generation sending [4].

Thus, this thesis project is focused on the analysis of the current state of SDN and its pros and cons for current networks, which can demonstrate some advantages of SDN after focusing on its complications, consequences, security risks and potential improvements [1]. We will add a simple controller, called sandbox.

Aims

The objectives of this thesis can be depicted as takes after:

- Investigation of the advantages of SDN contrasted with a conventional or traditional network.
- To propose a model to prevent DOS or DDOS attacks in SDN environment. That is, make use of Sandbox to prevent those attacks.
- Investigation in some utilization cases to eliminate the attacks related to SDN.
- Improvement of showing the model of sandbox to ease the traffic load on the network as showing the advantages of SDN.

Problem

Networks running under the SDN standard even now bring the same security necessities. Similarly, as conventional organize settings, as it will be probably that they will have a chance to be carried at times, private and secret data. SDN totally progressions those structural engineering and inter communicative parts of the parts in organize from this arises, a totally new [6]. Stage to attackers looking will perform security breaking attacks. This cause a need for comparable levels of security likewise, traditional network. Anyway, with guard against dangers of the separate network, this segment of the thesis analyses some of these key dangers, which means to defend their critics.

DOS attack, switch flow and table approach flooding, during those information plane levels, dishonestly made flow entries could be overflowed, and other units should be placed to expand the space in their stream entry tables [7]. This abandons those sending units unable to include the real stream records to their tables. This brings the answers in devices to being unable in including resulting to flow tables for updates and for divergent state. A standout amongst those key issues with those data planes units inside a product that characterized architectures is that of the switches inabilities with the separate between real stream solicitations also illegitimate ones. This imperfection considers attackers to perform effective DOS attacks at the information plane level toward filling those switches flow buffer for false importunity whilst it might a chance [8].

To be workable to target individual or distinctive data path also try to end its availability that the controller might a chance to be targeted effectively. Making and spreading a system-wide slip over accessibility. Those possibilities about this might a chance to be conceivably devastating, especially on processing settings, the place benefits perceiving secondary utilization will a chance to be unusable on customers and representatives. Furthermore, for accessibility to know customers removed, an adversary camwood arrange plane. To hold an attack which might point will trade off the confidentiality and integrity for delicate information on organizing for these motivations, those over DOS/DDOS attack have been said in this paper are viewed as amongst those the most majority essential attack sorts [9].

Methodology

Utilizing the functions of SDN, that permits a programmatic control over the streams. In the event of Identification from claiming malicious traffics, SDN might introduce packet sending standards for exchanging gadgets to the request should block those attacks [10].

A considerable analysis demonstrates that SDN, because of its centralized control will be unable that should give rapidly quarantine about compromise hosts. Also, a confirmation of validate hosts, requesting a remote confirmation dial to client administration server for user confirmation information, corrupting traffic or framework examining throughout enrolment. Furthermore, SDN is additionally fitted on gatherings to give immediate and grain control over networks, and acquires the chance will execute innovative security, also security strategies. Those fundamental clues will connect a virtual IP for every host to information transmission [1].

It may be changed unpredictably at the same time of the real IP location of the client is static. Controllers need to support the characterizing of those interpretations between the virtual IP and genuine up to preserve the integrity of the setup. Introduction of non-Flow administration planned on giving acceptable security that should create a privacy environment for clients. This methodology makes interpretation the middle of a non-ID, system IP and machine IP in view of SDN.

Main Contribution

In this suggested technique, we identify the usage and tactics of SDN according to DOS attack. Indicatively, over those creators display a low overhead strategy to movement investigation utilizing self-arranging maps to arrange streams [1].

This instrument is deployed looking into an SDN (NOX controller) system for enabling DOS attack identification. Our methodology contrasts from our recommended result not main performs DOS attack identification [2].

That detection also mitigation mechanism need support has been explained in this thesis. We intended to design a model to ease the traffic and to handle mitigation in our network, we called it a sandbox. However, the mitigate approaches utilizing a smart method that also empowers identification. Moreover, we utilize SDN standards so as to recognize detection and mitigate. The pernicious traffic starting from legitimate taking each user under consideration that those attackers generates real low rate traffic streams, also not simply to detect massive hitters. General abnormal traffic is distinguished in addition to keep focused on the multiple domains of local networks for the identification behaviour of mitigation [11].

Form of this thesis

This document is organized as follows:

- **Chapter 1:**

Definitions of Software Defined systems, its favourable circumstances and some essential SDN controllers.

- **Chapter 2:**

Shows SDN architecture and the differences between traditional IP networks and Software Defined Network. And what is OpenFlow.

- **Chapter 3:**

Shows DOS attack model and result upon idle time-out. And migration strategies according to the classification to install the same rules repeatedly.

- **Chapter 4:**

A proposal of another network is added, we called it Sandbox environment. We represent the model simulation on Mininet simulator.

- **Chapter 5:**

Showing the experiment process and its result.

- **Chapter 6:**

A brief idea about conclusion and future work.

Chapter 1: Software Defined Networks

Terminology

At the beginning of this thesis, definitions of important terminology used in this text should be clearly introduced. Main definitions are described below to differentiate between traditional network and SDN, OpenFlow, types of attacks of DOS and DDOS, and the idea of Sandbox.

A percentage of the protocols, administrations also provisions required secured compatible system environment for applications needed to depend on the clients, Furthermore need supports not perfect with those devices from other vendors. The point when the organizer may be arranged with make extended alternately new benefits would on make introduced, the existing foundation comprises of units starting with various vendors. The underlying framework needs to be modified; moreover vendors' reliance issue might cut off its arranged progress, also features and problems.

Traditional Networks

Regular networks administration advances suggested of the LAN (Local Area Network) also WAN (Wide Area Network), which would made from claiming different systems administration gadgets including routers, switches, also firewalls. A normal LAN interconnects an assembly from claiming group of hubs, generally in little geographic range, generally inside same fabricating for example, college, and home. Then WAN may be not certain idea to use in geographic area, in any case rather it camwood interconnected over noteworthy regions. For example, such that across the nation organize over a nation also it associate a large number of LANs together. In this thesis, growth for conventional systems administration innovations is restricted to LANs main [6].

Over LAN, information will be sent in the structure of packets, also different transmission innovations. Ethernet may be the particular case that will be a large portion. Generally, it may be used to specify in the IEEE 802.3 standard also its late form gigabit. Ethernet helps an information rate for 1 Gbit/sec, furthermore it substantially higher. A packet starts from source hub to achieve its destination hub, eventually switch perusing for receiving confirmation after packet reached the path. Toward those systems administration devices accessible in the network, like switches and routers.

Limitation of basic network

Since traditional networks changing traffic designs, improve cloud systems, and expand bandwidth to search for vendors solutions, but it didn’t meet those requirements to eliminate complexity [12].

Systems administration protocols need advanced in the time with convey enhanced reliability, security. Connectivity and performance, furthermore they need distinctive determinations and also similarity levels. Therefore, when progressions would wanted to a network, every last one of networks gadgets must make arranged with aggravate transforms under effect, coming about under generally static. Moreover, including a level about multifaceted nature on organize with succeed static nature, server. Virtualization is, no doubt used these days making networks progressive [13].

The expression virtualization need been around for a significant number a considerable length of time in PC (Personal Computer) science. In spite of it is an expansive term, like camwood a chance to be actualized in different layers of a PC framework alternately network, virtualization. Generally, suggests all the reflection in the middle of physical advantages and their legitimate representational. This definition from claiming virtualization will turn into clearer concerning illustration fundamental types for their improvements [14].

Previously, Virtual Machine (VM) movement acquires new tests to traditional networks for example, tending to schemes, directing based outline and forwarding. Also, all IP system is, no doubt worked will backing voice, information besides feature traffic, and looking after different Quality of Service (QoS) to separate provisions to each association or session expands the multifaceted nature of the group. Acknowledging the greater part these issues, an accepted system will be not capable to rapidly adjust on evolving provisions of client requests.

Acknowledging different QoS level service allocation, acceptable QoS approach must make actualized in those systems. Because of expanding versatile users, it will be not attainable for an administration driver should apply a reliable arrangement of the network, since it might aggravate those organize powerless to security no observances and different negative results.

A group must develop in line for those developing business sector requests on pick up economical. And aggressive markets, clients also benefits. Those system figure dissection might a chance to be. Helpful, Anyhow because of current dynamic business sector nature it doesn't give acceptable a great deal help to want versatility ahead of time. Those unpredictability of conflicting strategies connected ahead standard group limit the speedier adaptability of a network.

A percentage of the protocols, administrations also provisions required secured compatible system environment for applications needed to depend on the clients, Furthermore need supports not perfect with those devices from other vendors. The point when the organizer may be arranged with make extended alternately new benefits would on make introduced, the existing foundation comprises of units starting with various vendors. The underlying framework needs to be modified; moreover vendors' reliance issue might cut off its arranged progress, also features and problems.

Software Defining Network

By using decoupling the data plane and control plane, which is managing forwarding packets, Software Defined Networking (SDN) permits us to abstract the underlying infrastructure and applying the network. SDN will be relied upon on make a key authorizer for the following era networks, those represent 5G (5th era from claiming wireless or remote systems), which will addictive a reason to coordinate both IoT (Internet of Things) benefits together with traditional network [15].

In this chapter, after specifying the modern-day desires, I am able to narrate the ideas of SDN, its benefits and the diverse implementation.

SDN Controller

Those controllers based, OpenFlow authorized network structural engineering simplified the accumulated network logic through centralized or unified entity. Those controllers may be answerable for knowing the sending direction choices for forwarding and manipulate with also managing those flow tables on the OpenFlow switches. Once movement will be consistently sorted out done flows, its control gets less demanding moreover, considerably a greater amount is clear.

Chapter 2: Background

A prime past benefit in computer networks is the idea of Software-Defined Networking (SDN), which allows a harsh to enhance its behaviours browse centralized policies at a conceptually centralized harsh commanding. The SDN scheme replaces close down close, vertically integrated, and fixed function equipment on every side general purpose gather practical processing to criticize severely, usage scan openly, vendor neutral APIs by allocate software executing on centralized servers [6].

This undeceiving withstands exposes the capacities of irritating junk and provides chunk, generally concerning increased flexibility against matched of the SDN manipulation. The OpenFlow laws handle the conditions; duplicate an SDN importance delegation is highly difficult. Significant programming frameworks tracking either simple codes or current coded command specification of report and put up brief on ice for a contribution association between owner and the attack to get acknowledged, thereby enforcement corrupted authentication of complexity in SDN programming.

SDN represents a massive monologue when there is something unknown going on in standard network. The observant, SDN proposes an open architecture, in which the network carry on and change functions are decoupled. This architecture enables dynamic, straightforward supervise and conduct to procedure on significant servers click through a stable interface to dynamic applicable material. Fit criticizing carry out is not much longer care in same network light, SDN introduces an innovative is SDN controlled [1].

In this event, we evaluate SDN definition, its architecture, and its attacks and reintroduce prime aspects of the OpenFlow protocols, which enable controller switch interaction through a standardized protocol. In component, our analyses true approaches to the display of the irritating conduct or performance, more than ever, however software for OpenFlow networks and present solutions to DOS attacks on SDN.

Software Defined Networks Architecture

SDN proposes to assign an industry wide token of the relocation decision for the solutions and a methodical tradition review, distinguished method to learn the design and to cooperation nearly the forwarding circumstance. The forwarding abstraction and progress would permit advanced or exception methods, for example, routing algorithms, to be implemented on equipment, direct it is requiring a valid and delayed standardization proceeding. This determination promises to broadly speed faster of modernization in networks. The SDN sending plane reflection ought to make adaptable sufficient along these lines that an array. For group control calculations could make executed with it for containerize, it would serve to take on an operational level. It drifts just in the header packets based on Ethernet end apply oneself to, in return numerous applications may recruit to prepay on the insufferable of beginning addresses as significantly. Similar to one another, it would call endurable if the uncommunicative organization allows the routing fighting to moderate uncritical safeguard less ready to run off and tracking the random applications may stand by the know how to multicast [16].

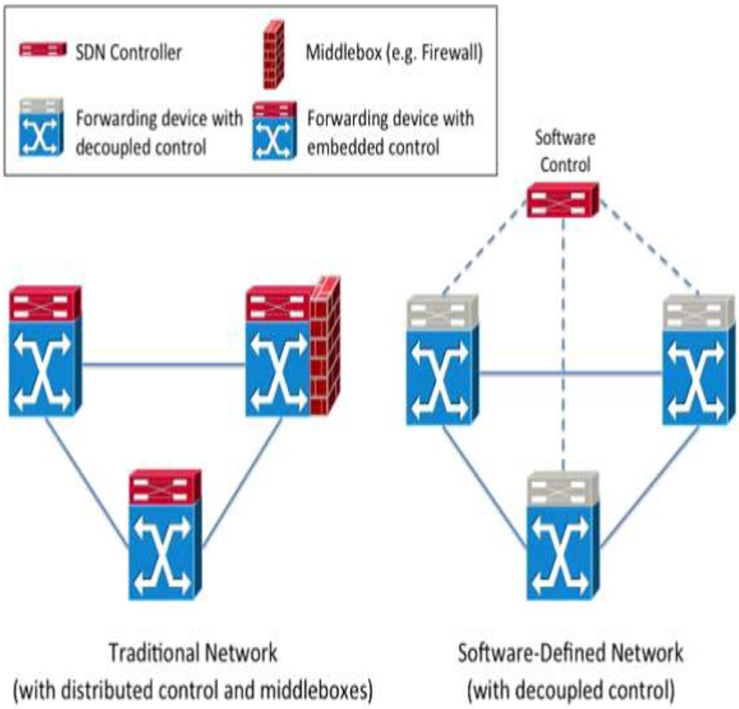

SDN Controller
Middlebox (e.g. Firewall)
Forwarding device with decoupled control
Forwarding device with embedded control
Software Control
Traditional Network
(with distributed control and middleboxes)
Software-Defined Network
(with decoupled control)

Figure 1. The difference between traditional IP networks and SDN [6].

The preface stimulates abstraction and meshing protocol networked setup protocols authorize further benefit: sending components are efficient toward uprooting those higher-level. Directing techniques are starting with those group components. Instead, the individual’s capacities could be a chance to actualized on standard computational elements, for example, such that product x86 servers that run directing calculations remotely, furthermore that use a standard controller-switch protocol. On design those sending component. These rearrangements might produce efficiency group components. Furthermore, easier convey for high octane system fittings. On addition, standard methodologies to building fault-tolerant administrations in information focuses might a chance to be applied to build the routing plane. Clearly, and only the sending task must be executed looking into every system component secure alongside. Common network architecture, those directing tasks will be also executed withal executed on each network. Component independently, despite the fact that each individual component might necessity majority of the data theorizes from different components in place will perform its part in effectively sending packets. The forwarding conceptual and protocol allows for a variety of approaches for implementing the electronic network control system [13].

Typically, this is done by having each network element broadcast all the local information that any other network elements may need in order to determine the influence state. For example, each network element may broadcast to all other network elements which neighbours it is directly connected to and the destination addresses of all directly connected end hosts [6].

Each network element, then independently calculates the desired route, determines its local role in forwarding packets along those routes, and then configures its local location accordingly. Since states can modify at any time, network elements broadcast updates to their local state and receive updates from other elements.

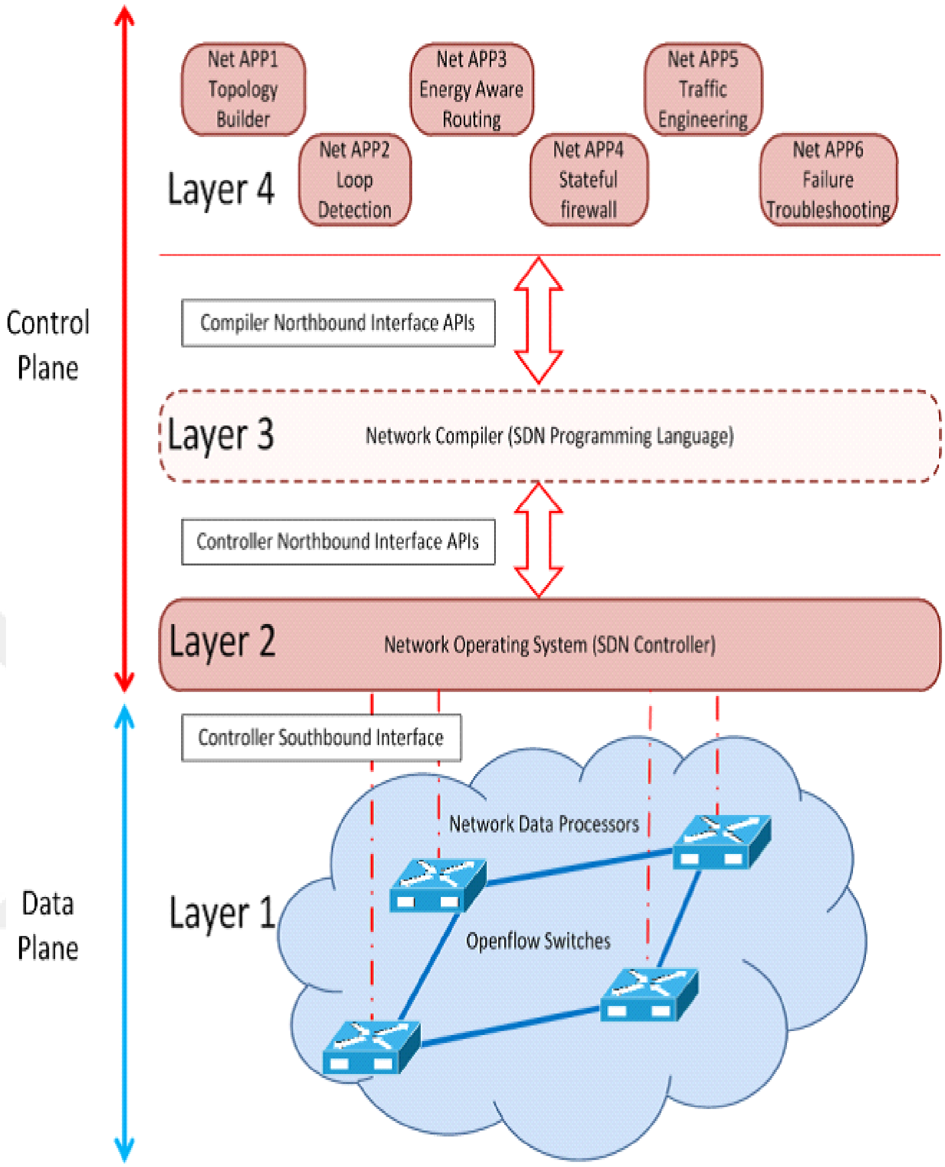


Figure 2. Layered architecture of an SDN network [17].

Detecting and mitigating attacks and intrusions is a major work in the field of computer networking. However, the focus was on the applications and host systems [18], [19], [12], and [16].

Some work in the field of protecting the network itself, especially for the trusted routing and DOS attacks on the network core was also introduced in [20] and [8].

In this work, the proposed mitigation model is based on the idea of Sandboxing. A sandbox is a method to force malicious activities to be performed in a controlled environment. The activities are monitored and classified to be legitimate or illegal. Legitimate activities are sent back to the working environment. Sandboxing was used in protecting computers and virtualization environments [13].

In this duplicator, a DOS attack targeting an OpenFlow switch is described, followed by a novel mitigation model to protect the switch against such an attack. The proposed model is discussed and shown to successfully protect the switch against the attacker.

OpenFlow

The OpenFlow standard has recently emerged the forwarding abstraction and switch-controller protocol. An OpenFlow switch initially establishes a communication channel over a TCP connection to a single remote controller at a specific IP address and TCP port. The OpenFlow protocol is then used to exchange information between controller and switch and to allow the controller to configure the switch’s flow table, which determines its forwarding behaviour.

OpenFlow is a protocol and a methodology of providing programmable networks. It had been originally proposed as a scientific experimentation platform enabling researchers to design and implement experimental protocols over an Ethernet switch with flow tables. The main idea in OpenFlow is to separate the forwarding algorithm (control plane) from the forwarding circuit (data plane) [4].

The architecture of OpenFlow is simple, as shown in figure 2; an Ethernet switch is used as the core component of the network. The switch is divided into 2 planes: the first one is a hardware switching fabric with a flow table that indicates a forwarding action to be taken. The forwarding (Switching) action is to copy the incoming frame from an input port to a specific output port.

In contrast to a classical Ethernet switch, where the switch flow table is built by a software running on the switch itself, (in Layer 2, or data link layer) in a plug and play using the Address Resolution Protocol (ARP). OpenFlow tables are built using the software running on the controller, they are more complex and a switch can have many flow tables grouped in a pipeline. The controller is usually import computer or a server, running an algorithm that builds the flow table(s) of the OpenFlow switch [22].

The OpenFlow controller is in charge of deciding how to handle switches without significant flow entities. In addition, it deals with the switch stream table by including and evacuating flow entities over the maintained channel, which utilize the OpenFlow protocol. The controller is in charge of keeping up all the system protocols, and strategies, broadcasting fitting guidelines to the network devices. The controller incorporates system knowledge. The switch must have the capacity to build up correspondence with a controller at a user configurable IP address to utilize the specified port. The switch starts a standard TCP association with the controller when it knows its IP address. In this method, the switch must recognize the approaching movement as neighbourhood before checking it against the flow tables because the OpenFlow channel does not travel through its pipeline. While, pipeline handling directions enable packets to be sent to subsequent tables for further handling and enable cluster data (metadata) to be conveyed between tables. With various controllers or single controller, the switch may build up connections.

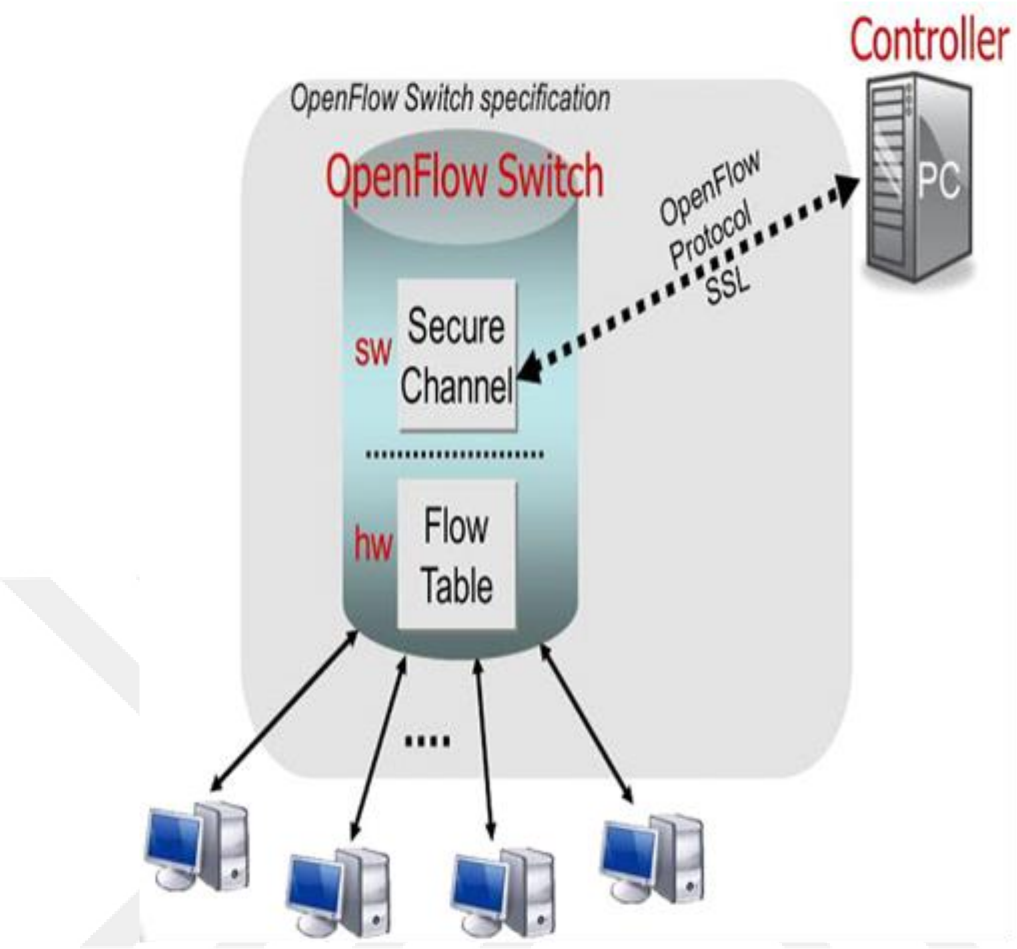
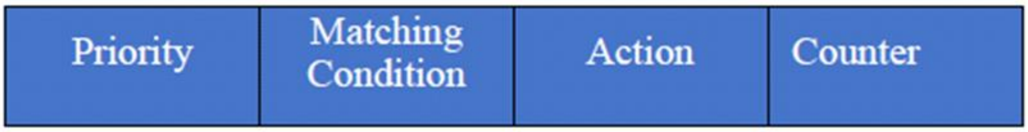


Figure 3. An OpenFlow platform [4].

In this layer, data processing devices exist. Data packets are handled based on the actions installed from the network controller on every individual device. As shown in figure 3, an OpenFlow rule consists of four main parts: priority, matching condition, action and related active counters [4].

Priority field is used to define the order of matching of a data packet against the set of installed rules; hence, once a higher priority rule is matched other lower priority rules are ignored. Matching condition consists of any combination of several IP/Ethernet headers (such as source IP, destination IP, port numbers, Mac address and VLAN id). The action field defines the action that should be taken upon receiving the packet such as forwarding the packet to an outgoing interface, continue processing in the flow table pipelines, or to modify some of the header fields before forwarding or to drop the packet. Finally, the counters field defines the associated counters to this rule such as the number of matching packets to this rule, its lifetime, and its last used timestamp and so on. Those values are sent to the controller to get some statistics about the data plane [8].



Priority	Matching Condition	Action	Counter

Figure 4. Structure of an OpenFlow rule [14].

On arrival of a new packet from the network, the OpenFlow switch process this frame as shown in figure 3. If a matching rule is found, then the frame is forwarded to the next flow table in the pipeline for further processing. A list of actions is updated. At the end of the pipeline, all the actions in the list are executed and hence the frame is either dropped, forwarded to an output port or sent to the controller. In many OpenFlow switch implementations, only one flow table is used instead of a pipeline. In case of no matching rule, the frame is sent to the controller with an appropriate state code and some statistics [19].

All rules in the flow table(s) are dynamically generated and deleted. In most cases, deletion is performed on an analytical basis where a timeout period is defined and if a rule exceeds this value without being used, it should be deleted. Figure 5 shows the operation of the OpenFlow controller and switch to process incoming data frames and manage the flow table(s) rule.

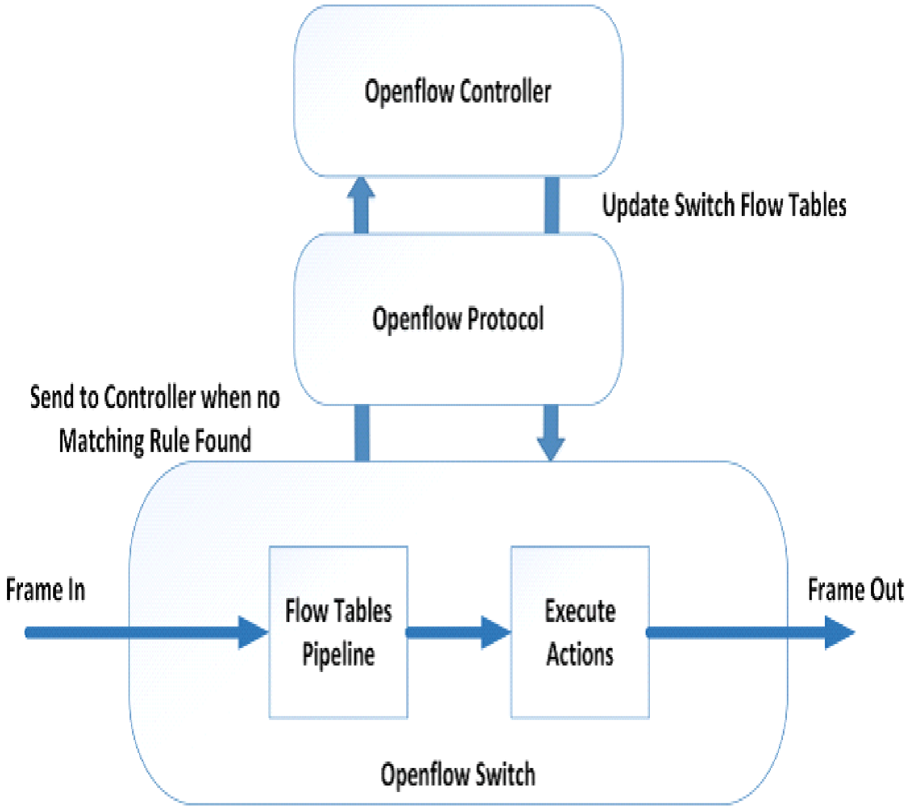


Figure 5. OpenFlow Switch Flow Diagram [4].

Attacks

The ping attack is capable to aim those hosts' to start with also simplest DOS attacks, in which those victimized person. Is overflowed with that's only the tip of the broken ICMP (ping) packets over it might handle. There are likewise those [3].

Ping for death attack, which timer frameworks might crash the point when they get, oversized ICMP packets [3].

On SYN Flood attack, those striking frameworks sends SYN messages of the utilized person server. System, which show up should be real anyhow indeed reference a customer framework that is. Unabated should react of the SYN/ACK messages. This indicates that the last ACK message will never be sent to the target server structure.

Information structure of the utilized user server framework will automatically fail, providing it for immoderate on accepting any new approaching links [3].

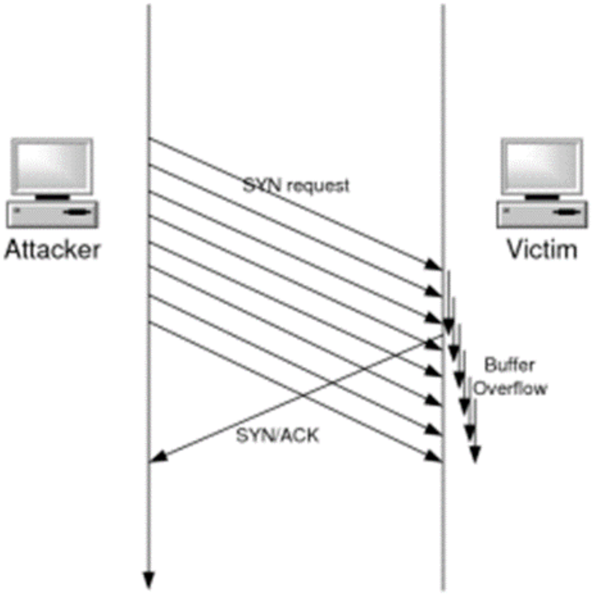


Figure 6. SYN Flood attack: The victim receives an overwhelming number of illegitimate links Request [3].

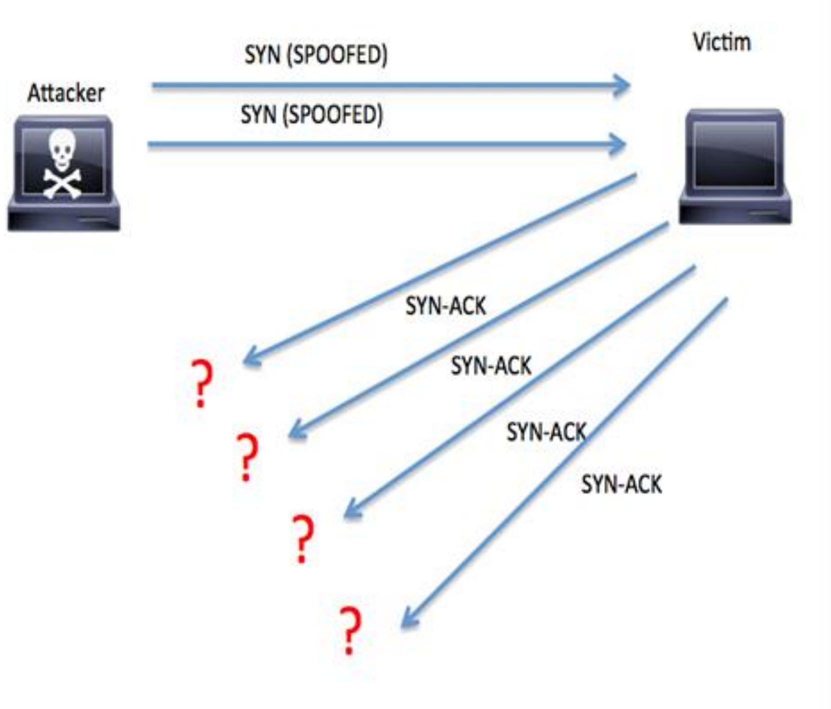


Figure 7. TCP SYN flood attacker against victim mitigation [23].

Those representatives which makes following of the DDOS link packets on a scale were troublesome on need significant volumes for movement, making it simpler for that attacker with hidden neighbourhood from group director. Those attackers are never again needs to support a rundown about representatives, since basically log on to the server and perceive a rundown for every last bit accessible operator [7].

The process programming introduced in communicates for different type of channels, additionally notifies the attacker when the process is dependent upon the networks. Likewise, it doesn't give a difficult record for sharing, which is an action amid the indifferent techniques of process code distribution. This makes it less demanding to attackers on secure additional, utilized people will consume for their attack [11].

In the Smurf attacker, the attacker will be utilizing ICMP effect ping packets guided on IP show addresses starting with remote areas on producing DOS attack. There are three assemblies for these attacks: the attacker, the trafficker, and the victimized person [24].

The user receives an ICMP effect request packet guided of the IP show location claiming from their organization. On the channel ICMP movement guided, I should show addresses, huge numbers of the machines on the group will accept this ICMP effect demand packet, what’s more send an ICMP effect answer cooperation packet again. When every last one of the machines looking into a system reacts will this ICMP affect tracking the outcome, which might make extreme group delay and undetermined interruption when those attackers make these packets, they don't utilize the IP deliver of their own machine. Concerning analogy the hotspot address instead, they make frame packets that hold numerous those spoofed location of the attacker’s exceptional utilized person [25].

Those consequences are that at every last one of the machines toward. Those traffickers site react of the ICMP effect tracking, they sent replies to the victim’s machine. Those victimized people will be subjected will group delay might be feasible. To resolve and classify the attack, which employments UDP packets as against to ICMP affect packets.

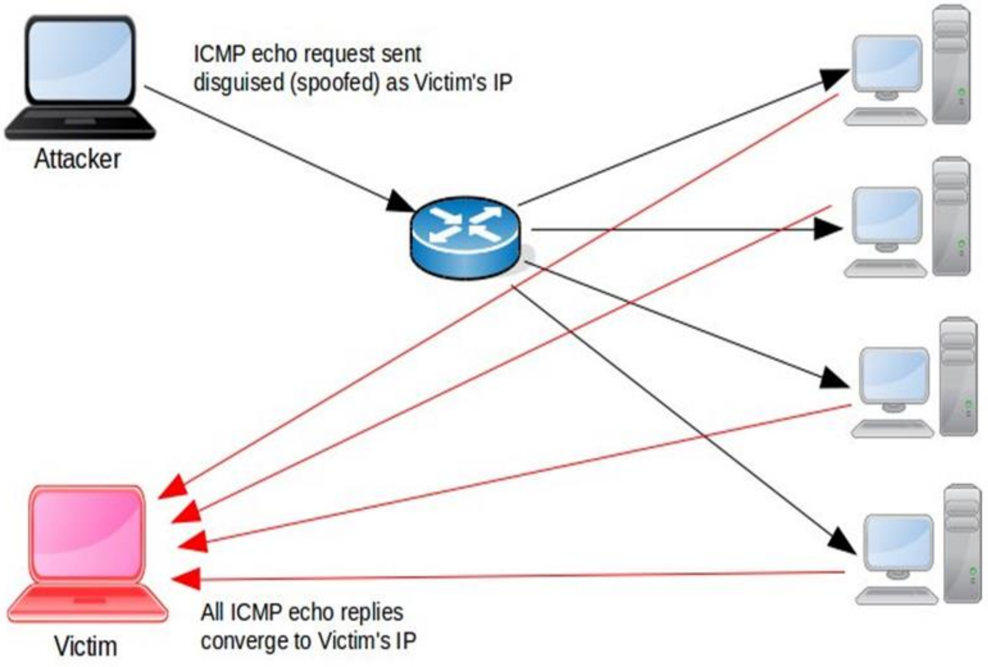


Figure 8. . Smurf Attack. The attacker sends a large scale of ICMP traffic to a program destination and uses a victim’s Information processing address as the source Informatics so that the replies from all the twist that respond to the broadcast address will flood [24].

Teardrop attack misapplies those ways that the Internet Protocol (IP) requires a discontinuity for a packet that is excessively less for that next switch to handle those divided packets, identifies a counterbalance of the start of the initial packet that authorizes the whole packet should be a chance to be reassembled toward in the assigned framework. In the harmful attack, those attackers puts a confusing worth in the second or later interval, moreover assuming that the accepting working framework can't adapt to such disconnection, then it will probably crash. Unusually, another mixed capture about harmful emerged bonk, which functioned particularly for a trick in this update [24].

Bot armed forces. A few aggregations for cyber-criminals practice previously, traffic huge numbers of workstations defenceless should deny the administration to attack [24].

They produce these huge armies from handling tasks, of a couple of thousand with supposedly dependent upon 1.5 million computers, and lease them to the possibility DOS attackers.

DENIAL OF SERVICE ATTACKS

Done February 2000, the sites of Yahoo, eBay, Amazon, Datek, Buy, CNN, ETrade. ZDNet Furthermore dell was "around those focuses on a 15-year old Canada wild rye nicknamed “Mafi-Aboy”. The attacks, which arrived at those rates from claiming 1GB/s wreaked immeasurable people in the victims’. Economies transformed those lifestyle people consider DOS. After that incident, DOS began operations for amplification attack for hacking differentiation on financial crime, cyber-warfare. And politically stimulation, attacks some pending requests, later the frameworks will focus on this issue from claiming [11].

The port forward for handle business sectors around the world, and the DNS root servers of the Web 3. R.T. Morris cooperation that the weakness in this scheme is the IP protocol is that the source host itself fills in the IP source host id, and there is no provision to discover the true origin of the packet [26].

This may be the point when it got to be reasonable that relying upon the determination of the attacker. Practically each workstation system administration could a chance to be disturbed by a few sorts DOS attacks. Furthermore, frequently for long periods of occasion when the accompanying would damage those computer networks in practically of every service can be of the new sorts of attack that showed up throughout this period. The Distributed reflector DOS attack will be fundamentally the same over the idea of the “Smurf” trap [23].

The attacker Request as much armed force for compromised workstations should send association with a few superbly real computers, requests yet all the utilizing the victim’s IP for their packets when the real machines answer cooperation with these requests from the collector from stating constantly on is those victimized people. The coming about DOS trick will be all the more distributed, that's only the extra standard tip of the broken. Also, fundamentally, all the more challenging should follow. A Denial of Service (DOS) attacker may attempt to perform DOS attacker to the controller or use different intends to bring about the controller to come up short. It’s any intended effort to prevent legitimate to utilize from reaching a specific mesh network resource [23].

For instance, attackers may attempt a few types of recommendation to utilization attackers on the controller to steal it, cause it to react to a great degree gradually to approaching packets, and make it moderate to send messages out. Shin and GU exhibited a doable and viable DOS attacker to SDN systems, which contains two stages to uniquely mark whether a given system utilizes SDN/OF switches. The observable verification of an SDN switch relies on the recognition that the response times for receiving the packet, may be particularly incorporated to identifications. Since the stream setup time to create the new flow that made stream demands from data plane to control plane [2].

Denial of Service attack is whatever proposed should seek to block real clients starting with arriving at a particular group of attackers. A decade later it got clear that the attackers might routinely misapply this shortcoming of the IP. Eventually Tom's perusing faking their source address, an act which is called IP spoofing. A standard nonspecific DOS trick is practically continuous broadcast (DDOS) Distributed DOS attack [11].

Those attackers' takes control of an immeasurable number from claiming particular ensured computers, for example, the individuals. Without firewall also, up and coming antivirus software, furthermore request them to send at the same time. Volumes are a negligible movement with a particular focus concerning the analogous result of a chunk. Routers gains in the region of the focus would overwhelm and a number about real customers can't unite with it any longer. The standard focuses would those servers about e-commerce. Web sites, which camwood fair huge budgetary misfortunes. Other focuses might a chance to be news websites. Corporate networks, banks, the national infrastructure, market competitors, and so on.

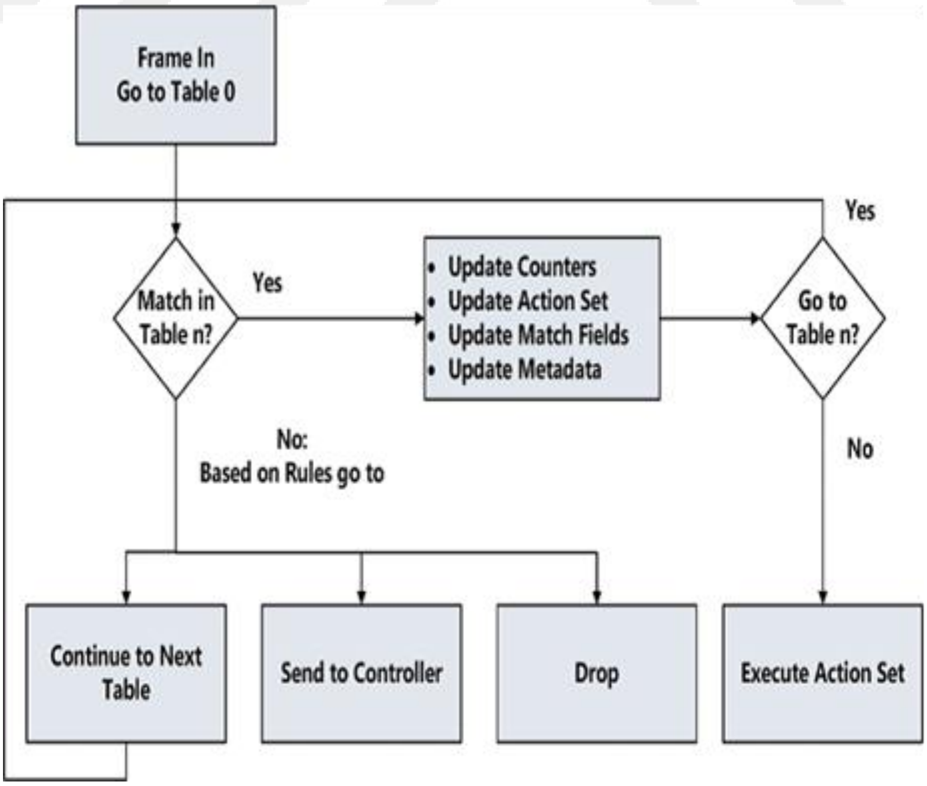


Figure 9. OpenFlow switch and controller processing [2].

This kind of attack is a significant risk for OpenFlow based SDN, as it can damage the entire system, for example, corrupting system execution, dropping packets. Likewise, as OpenFlow controller is a key segment for system control and administration, it has turned into a bottleneck and is incredibly weaken by such attackers [27].

Dependable encryption can't secure SDN from being attacked; however improved trust components and access approaches can moderate such attackers. At the switch level, it is alluring to have an ability to consider stream table principles, guaranteeing an administration to goodness has and denying unapproved access. Current OpenFlow switches have not actualized such knowledge while additional overhead is the significant concern.

Overall, an attractive security system ought to guarantee both the controller and the switches can rapidly regain from huge volume movement attackers. Legitimate checking and reaction systems are relied upon to be dynamic when identifying DOS attackers. Case in point, Shirali-Shahreza and Ganjali proposed FleXam, an adaptable testing expansion for OpenFlow that authorizes the controller get to packet level data, which can distinguish DOS attackers [26].

Additionally, stream data are helpful in identifying most DOS attackers, through breaking down activity. For instance, stream headers can be utilized to distinguish DOS attackers if there is an imbalance among approaching and active movement.

DDOS attackers can be caused to flow the connection between the controller and the system in order to keep the controller from handling streams. To avoid such attacker, the controller ought to be put inside the border, and streams to the controller should be cut on oversaw interfaces on the off chance that they are not from OpenFlow device. Along these lines, the controller can be secured from flooding attacker [11].

Existing Solutions to DOS Attacks on SDN’s

At least from claiming Section 2, we included identification similarly as the to begin with a fundamental component from declaring this is not actually exact identification. That might not to be required in the perfect gas body of evidence and a resistance, structural engineering for proactive qualities, which might provide incomprehensible at whatever DOS attack over proactive architectures. Have been proposed, a couple from claiming which present after the fact that they do have a chunk as a repeatable point. Hindrances separated starting with the truth that to date no framework may be perfect. A DOS attack against one’s system doesn't happen precise often and in any event resource-wise. A proactive insurance framework may be normally excessively reasonable to work in the nonattendance of an attacker. This might be a chance to be avoided though that security framework begins operating straight after a possibility. The attacker will be detected, which provides the identification, progress essential secure alongside practically sensible instances [5].

With those perceptive furthermore programmed arrangements that might offer, would progressively be utilized within DOS identification of a key issue, it secures alongside utilizing this population of systems. That choice allocates the entered features that will assemble to give suitable information. What's more significant majority of the data over those approaching movement. Additionally, these frameworks must have a chance to be intended to make quick sufficient adaptation to the precondition of ongoing rules [28].

A few routines are recommended for actively testing the clients’ legitimacy. For Netbouncer continuously assign tests are needed, as to keep a rundown in order for clients to be beyond suspicion. At the same time experience of systems left require arrangements for testing isolated packet, under based ICMP for analogy. A basic ICMP affects the request, flow-based originally for stateless TCP/SYN cookies. Furthermore, transport protocol (such similarly as CAPTCHAs) has a different capacity of administration strategies that would be used to guarantee a reasonable solution with the attacks. Eventually, Tom's block the process of the real clients, at the same time this authenticity expires and then after specific interval, the fundamentals should be tested once more [29].

In spite of, the larger part of regular cases use all the capacities for grouping, mishandling personalities of real customers could handle this sort of resistance, a comparison of particular idea is investigated in the resistance results which need cooperation to be dependent upon complications.

Over connection design DOS attack, the customers need operational approach, taking care of minimum cryptographic to mystify in front of their association request to be ordered. It might take less time to solve and to protect the server, which it will be to a large extending quicker in checking results. That can back off the attacker, yet it's not assuring enough, since mislead generates a transformation, which could be allowed through the attacker’s rate [23].

Analogies of such methodologies could be discovered secure additionally; exactness of the over dynamic test results might accomplish great levels about identification of accuracy, as starting with those regular deficiency of continuously utilizable as DOS base. They are in the same method concerning analogy the simplest type of dynamic validation, those ACKs, would utilize likewise DOS holder in the reflector DOS attack. Kim has recommended information about the data mining approach dependent upon a programmed characteristic. A determination component with a neural system classifier for DOS attack identification [5].

They used a decision tree, together with entropy, chi-square concepts, will select those best to allocate the nomination offers of these features that need cooperation to be utilized within the neural system classifier [16].

Previously, their experiments, the hopeful qualities were those octet check for every flow, the packet check for every stream (P/F), those tcp/ip octet number for every flow, the tcp/ip packet check for every flow, those. UDP octet number for every flow, those UDP packet check for every stream, the destination port difference to Tcp/ip movement (srcTport), the source port difference to UDP traffic, the end port difference to tcp/ip traffic, those end port difference to UDP traffic, those source ip deliver variance, those TCP/IP movement pro-chunk (T ratio), and the UDP movement pro-chunk that choice [19].

The decision tree P/F what’s more search-port for utilizing both the entropy and the chi-square methodologies. Accordingly, the majority of data regarding to the grouping classification of TCP, UDP are an inauthentic mixture of the two, might have been utilized in the choice mechanism, P/F, ratio what’s more srcTport were the naturally decided qualities [16], [22].

Kim utilized a versatile Neuro-Fuzzy Inference frameworks (ANFIS) together with the Fuzzy C-Means Clustering Algorithm (FCM) on recognize DOS attack. Also, Kim tried their strategy toward performing examinations looking into a DARPA/KDD99 dataset Lee et al suggested plane the main place, those substance of the approaching packets are analysed, after the analysis, identification framework utilizing fuzzy cognitive maps (PDSuF) examine detection system using fuzzy, is used to identify

DOS attack, furthermore a dark analysis about IP addresses will be constructed to secure alongside of understanding the gathered data [30].

Mukkamala and Sung connected three computational perceptive techniques support vector machines (SVMs), multivariate adaptive regression splines (MARS) and linear genetic programs (LGP), for a multi-agent setting of the issue about DOS identification, also compared the test outcomes got toward the sum three techniques. To these perceptive methods, Sung also Mukkamala, additionally tended to those characteristic choice issues and provided for characteristic positioning calculations for maintaining performance reason of investigation by joining DOS attacker analysis. Additionally, system protocol Mukkamala have utilized various classifier frameworks for interruption identification, moreover, reports individual outcomes for DOS identification. Chan suggested entropy based characteristic grouping system for different classifier frameworks also utilized RBFNNs concerning analogy base classifiers [31]. Shin arranged to minimize false alarms in DOS identification. Eventually, Tom's perusing for utilizing a decision tree approach [16].

Conclusion

Cetnarowicz and Rojek also apply a multi-agent methodology of the PC security issue stress first DOS attacks, particularly, by which they separate the traffic to the point on the exact flooding area of the virus [32].

Furthermore, terrible self-destructive considerations and conduct in utilizing ideas to start living in a secure framework. Seo and Cho speculate based plan from a Black Board Architecture (BBA) that firewalls should recognize DOS attack, also structure to monitor IP addresses [5].

We totally array the DOS attacker identification in the network, which suggested large chunks about these are considering of typical analysis of the movement packets. Moreover, specific IP addresses of other critical data about the packets different methodologies that would consider the timing aspects of the packets flows. The larger part of maintaining them require or expect exactly representational about the thing that will be normal movement stream concerning the analogy contradicted, showed DOS related flows. Massive numbers of the systems require a connection numbering or taking in, period that is used to make statistics, information improve facts that camwood make used to analyse with presumed attack. Despite each from declaring these methodologies offers a few extremely absorbing thoughts, also insights under DOS attacks little assuming that at whatever load in, need to be conveyed out as such with consideration at these separate methodologies against one another. In addition, these methodologies deliver comparative sorts about the attacks; there will be likewise a critical differentiation of the actual attacks that these different identification schemes would attempt to address.

Chapter 3: Denial of Service against Attacks OpenFlow Switches

DENIAL OF SERVICE ATTACK MODEL

In [23] and [29], the authors introduced an attack targeting the OpenFlow switch flow tables. The attack fills up the forwarding rules memory. An OpenFlow switch usually has a fixed size memory to store its forwarding rule list. The memory size is limited to N rules where N varies according to the switch brand. For example, in HP 5406zl switch N=1500 [2], while in CpQD's OpenFlow 1.3 Software Switch [25] N=4096.

If the flow table memory of the OpenFlow switch is full and the controller instructs the switch to install a new flow rule [12], the switch detects that its flow table is full. As the switch cannot install this rule, it sends an OFPT_ERROR message to the controller with error code OFPFMFC_TABLE_FULL it then drops this packet. The switch cannot forward buffered packets until there is space in the flow table to install new flow rules.

Attack Environment

Figure 7 shows the attack environment. A simple network with one OpenFlow switch (S1), one Controller (C0) and two host (h1 and h2). Host-1 (h1) will act as the attacker targeting the switch (S1).

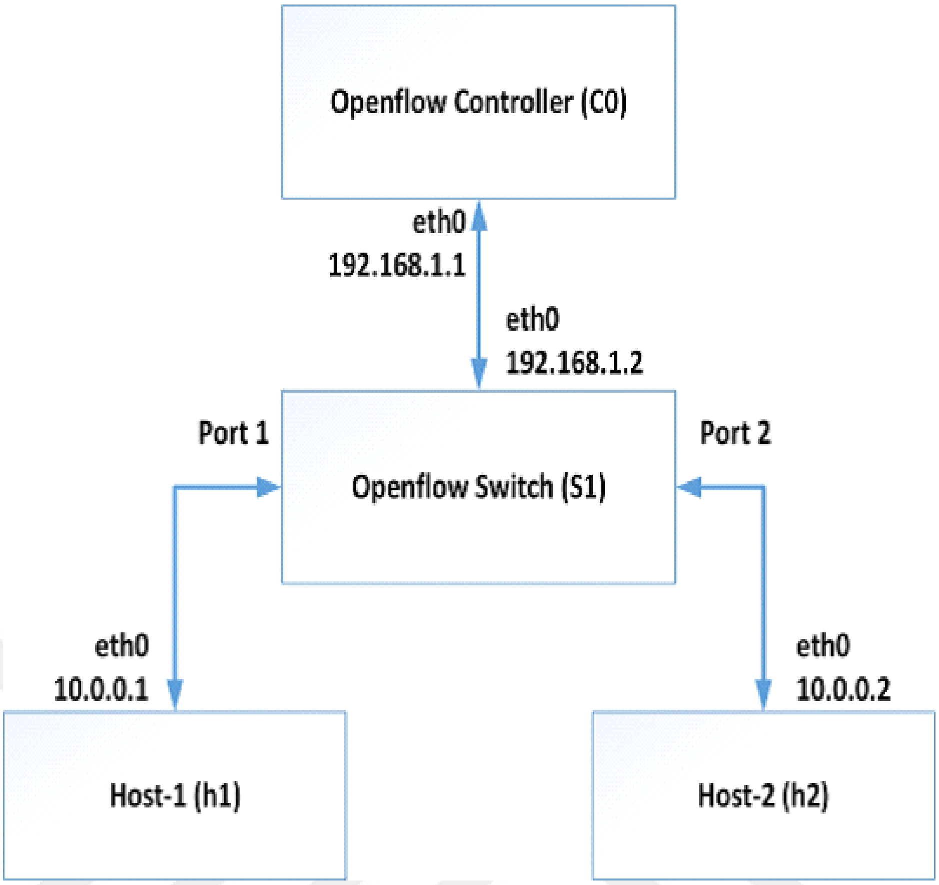


Figure 10. Emulation Topology [23].

The attacker at (h1) attacks the flow table of (S1) by sending streams of UDP packets with the destination address of (h2). This means the attacker knows at least one host IP address in the network. Those UDP packets are constructed such that each arriving packet will force the switch (S1) to forward it to the controller (C0) making the controller (C0) to install a new Rule in the switch (S1). The main utilize here is to change the source port number to a new value (sequence) in each sent packet. The traffic is generated using a hi-ping. A total of 50000 UDP packets (with 512 bytes of payload) at a rate of about 1000 packets per second is generated and sent as described in [23].

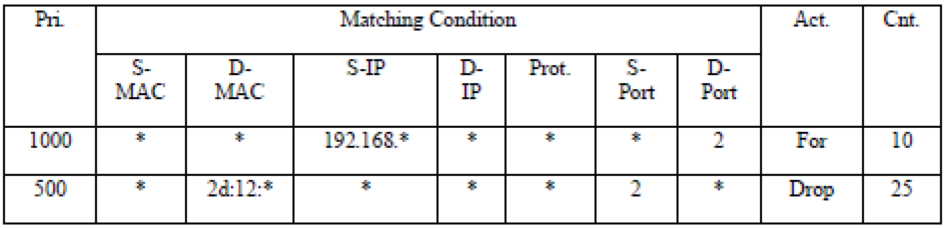
Refer to figure 4, the Matching Condition part of the rule is a tuple of values extracted from the packet header. Typically, this tuple contains the source MAC address, destination MAC address, source IP address, destination IP address, protocol, source port and destination port. Note that the MAC and port values are those used in classical switches for their ARP table. In addition, wild cards like (*) are used in the flow rule to indicate a field needn't be matched.

Attack Results

Every stream entrance needs an idle timeout What's more a hard timeout connected with it, both from claiming which would arranged through the OpenFlow controller. Those idle timeout may be that scale from claiming seconds, after which a stream entrance is uprooted from those stream tables and the equipment enhanced in view no packets match it [6].

Those difficult timeout will be those numbers of seconds following which the stream passage is evacuated from those stream tables and the fittings if or not packets match it. In a stream passage need both an idle timer. Furthermore a difficult time connected with it, those initial timers with lapse makes the stream entrance will make uprooted. On those idle timer expires in those stream entrances will be uprooted at that purpose just if there are no matching packets. Otherwise, the stream entrance will be uprooted at the difficult timer expires.

The results in [23] show that the flow table of (S1) will be flooded by the rules, making new arrival packets to be dropped and the switch (S1) effectively out of services.



Pri.	Matching Condition							Act.	Cnt.
	S-MAC	D-MAC	S-IP	D-IP	Prot.	S-Port	D-Port		
1000	*	*	192.168.*	*	*	*	2	For	10
500	*	2d:12:*	*	*	*	2	*	Drop	25

Figure 11. Show an example flow table [29].

One factor affecting the switch (S1) behaviour is the rules time-out pre-set value. OpenFlow switches delete any unused rule, if it is not used for a specific time-out value. If the time-out is small enough, unused rules will be deleted more rapid lowering the effect of the attack, with the side-effect of huge traffic between the controller and switch(s) to install the same rules repeatedly. On the other hand, a larger time-out reduces the controller-switch flow, but makes the switch more vulnerable to flow table floods.

The packet drop rate is measured as an indicator for the successfulness of the attack. As expected, with larger idle timeout values, previously installed flow rules remain in the switch's flow table longer. This causes the flow table to be filled up and thereby prevents new flows from being created [23].

Conclusion

Migration strategies try to arrange nodes according to the classification must think as the attacker might utilization for misclassification should their relating to point. Orders schemes ought to further strengthen represent information passing lost packets in the network likewise with the data could be allowed. Relief advance makes mitigation technologies effectively joined, just when they support data or when they are planning on not to send to one another. Generally joining frameworks might present vulnerabilities, improve, expand attack acceptability.

Chapter 4: Protecting OpenFlow Switches against Denial of Service Attacks

PROPOSED DOS ATTACK MITIGATION MODEL

Denial-of-service (DOS) attack would on the climb and need developed under complex, also certain security tests associate extensive with also little despite dos attack need assistance, not a late phenomenon those routines. What's more assets accessible will find the behaviour of attack that have such dramatically advanced with incorporate distribute (DDOS) [11].

DOS attack consumes assets done networks. Server clusters, improve end hosts for those dangerous destination for keeping in review that administration extremely devaluing on real clients' assets that are regularly expended previously, such attack incorporate system bandwidth. Server improve switch CPU cycles, server intrude preparing capacity [33]. In more particular, protocol information structures analogy dos attack. Incorporate TCP/IP SYN attack that consumes protocol information structures around. The server working system; ICMP guided broadcasts that regulate shows deliver on send a surge for ICMP replies to a target group.

Distributed Denial of Service (DDOS) attack upset. Furthermore deny real workstation to be more group asset. Use through compromised hosts that consume attacks. Relief innovations need been produced. Should protect against DDOS attacks; be that there may be little comprehension of the key associations the middle of DDOS attacks, relief strategies. Also attack execution without a strong comprehension for these basic relationships, it may be was troublesome with determine the capability of relief innovations on location those DDOS issue or how relief advances might effectively make deployed together [10].

MAC protocols work at the join layer and a large segment require participation. Between hubs on referee channel use, making them especially defenceless Dos attack. Link-layer dangers incorporate collisions, investigation, also packet recharge. An impact trick is synonymous with those reactive-jamming attacks [7]. That could relieve exactly collisions by utilizing error-correcting codes. An attacker may decide will execute denial-of-sleep attack, through basic Jamming-based dos attacker on a WSN should cut off those attack's span to mask and conduct. Incapacitate a sensor network, insert a trick might devaluate months to drain those focused device's batteries.

MAC protocols need cooperation a characteristic keep tabs to denial of sleep attack. This may be, a result they control those purpose of the transceiver, this expends greater a scale vitality over whatever viable part ahead on most wireless sensor platforms. Those join layer coordinates right of the physical medium linking network's hubs.

Different physical layer attack incorporates hub adjust or decimation. Although, it's hard to prevent the destruction of Hubs deployed over an unsecured area to excess hubs and camouflaging camwood. Relieve this danger defences against. Improving incorporates nodes, tamper-proofing packages, or actualizing improves response likewise cryptographic memory. In link/MAC layer, MAC protocols work during those connection layers, also MAC protocols require collaboration [6].

The middle of hubs with referee channel use will make them especially powerless dos attack. Link-layer dangers incorporate collisions, examination; also packet recharge impact trick may be synonymous for those reactive jamming attackers to might relieve percentage of collisions by utilizing error-correcting codes. However, ECCs Error Checking and Correcting includes transmission overhead expending extra vitality [7].

A duration attack utilizes the two-way are request-to-send and clear-to-send (RTS/CTS) handshake that numerous Mac protocol use to relieve or mitigate. Eventually Tom's run after the hidden node issues an attacker might weaken a node's to benefit over sending RTS messages will inspire CTS reactions starting with a focused on the neighbour's hub. Furthermore, hard link-layer confirmation might relieve these attacks. However, a focused on hub getting those fake RTS. Messages still expend vitality, and group transfer speed Routing-disruption attack camwood head of DOS attack in multi-hop sensor networks [29].

Vulnerabilities attack countermeasures general attack once directing. Protocols incorporate spoofing, replaying, or adjusting directing movement Link-layer. It authentication or confirmation and entire play might sufficiently keep these attacks detrimental hubs that destructs the network's directing protocol might scale. DOS trick towards making itself and only huge numbers, routes dropping the greater part of packets in the black hole attack, it specifically ahead packets with decrease the likelihood of identification [24].

Successful relief about DOS attack is a compressed environment because its pace in a minimum rate issue on the internet in frequent cases, DOS attack a chance to be those event of a possible significant occasion. Administration retracing, vented assuming that the spoofed hotspot IP address will be followed again to its root, which secluding those TCP. SYN packet must a chance to be transformed on learning its legitimacy [7].

Missed hosts, furthermore domains starting with remains of those groups. Recently, IP trace to despite the fact that the asset use connected with the process back. Forward considering of probabilistic packet marking (PPM) has been recommended to accomplishing trace back of DOS attack. In this duplicator, we demonstrate those probabilistic packets marking of interests because of its effectiveness what's more implement ability. Vis-a-vis deterministic packet marking the logging or informing based on schemes-suffers, under spoofing of the denoting field in the IP header by the victim. It might show that there may be an exchange off the main capacity of the user with confines those victims and the seriousness of dos attack [34].

Likewise, it may be represented a capacity of the marking probability, length, movement and volume. The ideal solution of PPM can indicate the probability of those attackers, since it assigns them from spoofing and inspect the estimate source address. Ability to be communicated concerning analogy, a force of minimal streamlining problem for the assigned victims. To check feasibility of the scale to create tactic ways for minimization. Under conveyed DOS attacks, those traceable influence achievable PPM that perusing attacks could make amplified, which reduce the acceptability.

In spite of proactive server roaming extinguish principal progress to the relief about dos attacks, its impediments. Would tend to in 5 processes:

- It handles a special case in the server, dynamically during a time.
- It requires logging off administration, subscription, which will be not an adaptable administration model.
- Servers keep track of the IP addresses about every one subscribed customers over. Request to occasionally send them roaming upgrade messages; Keeping such a

rundown maybe not scalable, particularly for an expansion. Also, customer amount may be cut off to use upgrade IP address, decreasing flexibility.

- The component will be not transparent to the customer that requires transforms to customer software.
- It will be not difficult will trade off a customer machine for probability about possibly uncovering the administration of members details saved in the customer improve monitoring for client’s movement with uncovering the delivery of current server or for both.

TCP migrate also migratory TCP, which provide a structure to operational person end point of a carry on with TCP/IP. With one area, merger start area hosting and improve IP location or an improve port number, need cooperation to be utilized for adaptability to attack any tolerance in the network spontaneity administrations will be a framework to consider. Reactively, relocating the administration front-ends and updated main plane and assembles it for merger relocation the traffic, also gives a secure structure to issuing the roaming trigger proactively [20].

This idea about introducing a layer about indirection should preserve against DOS attack might have been exhibited in the Secure Overlay Services (SOS) building design and the DOS Attack Mitigation (DAM). SOS employments the overlay network about proxy hubs to conceal the areas of little number. Also permits main movement from these servers with enters the security services group in place should add an entry of the overlay network, a customer need will validate itself. With a standout amongst one of those replicated access points (SOAPs). Which, Routes every customer packet should a standout amongst those servers utilizing a hash based. Directing this overhead of overlay direct routing to make the dependent servers upon 10 times those immediate correspondences the inactivity. Anyway employments one-hop tunnelling through the AGs typically, close in neighbourhoods with clients as against to hash-based directing for client-server traffic; subsequently it abstains from these high latencies [10].

The idea of Sandboxing is utilized to protect OpenFlow switch against the DOS attack of the same type described in chapter 2. The following subsection describes and analyses the proposed model.

OpenFlow Sandbox

As shown in figure 12, the same network utilized in the previously described attack is used again. It is the left part of the figure marked as "Operational Environment". Another network is added and marked as "Sandbox Environment". To virtualize more in reality the result of the Sandbox with SDN for the first time as it is a unique idea.

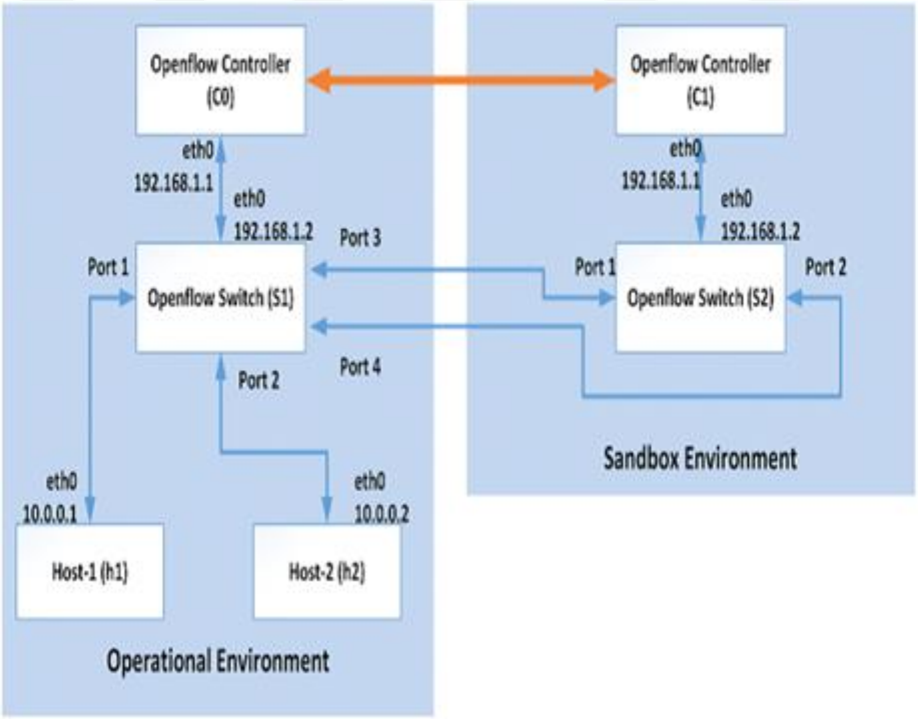


Figure 12. OpenFlow Switch with Sandbox.

The two networks are connected together on 2 levels.

Switch (S1), the target of the attack, has now 2 more ports marked 3 and 4. Those ports are used to connect the switch to the sandbox switch (S2) to its ports 1 and 2.

Also the two controllers (C0) and (C1) have their own protected communication link using SSL and public key certificates.

Initial Configuration of flow tables

The operational switch (S1) is initialized with 4 low priority rules that directs all traffic from port 1 to port 3 and vice versa, and the same for port 2 and 4 as shown below.

The rules tuple, refer to figure 8, will look like:

- 1, *, *, *, *, *, 1, *, forward to port 3, ---
- 99999, *, *, *, *, *, 3, *, forward to port 1, ---
- 1, *, *, *, *, *, 2, *, forward to port 4, ---
- 99999, *, *, *, *, *, 4, *, forward to port 2, ---

The rules will cause any traffic coming on port 1 or 2 without a matching rule; this flow includes the attack, to be forwarded to switch (S2). The priority level of those rules (first and third) are set to (1) minimum so they are executed only if no other rule is matched.

Also the second and fourth rule is given a very high priority level (99999) so they are executed before any other rules.

Switch (S2) tables are left to be configured by its controller (C1).

The time-out value of switch (S2) is set to a small value, so rules in its flow table are purged rapidly.

Communication between the controllers

The two controllers (C0) and (C1) are connected with an out of band secure channel; they communicate dynamically during the operation as flows:

From (C0) to (C1): Periodically (C0) reads the flow table of (S1) and send any rules with port 1 or port 2 as final destination to (C1), where (C1) orders (S2) to implement the exact same rules. This action will make (S2) to look exactly as if it is a 2- ports switch connected to the hosts (h1) and (h2) of the operational network.

From (C1) to (C0): After each N time-outs intervals of (S2), the controller (C1) will read the rules in (S2) flow table, deletes those rules from (S2) flow tables and finally send them to (C0). Immediately, (C0) will implement these rules on (S1) flow table. As those rules are not deleted after N time-outs, they can be classified as safe rules.

The parameter N is adjusted for performance, a value of 0 makes no Sandbox. The smaller N saves some time as lower volumes of traffic will be sent to the sandbox, while a larger N offers more protection.

Analysis of the proposed model

The proposal will make the Sandboxing network to get all the new traffic from the operational network if the packets are new and do not have a matching rule, creates prober rules for those new packets on the sandbox switch (S2) and sends the traffic back to the operational network for normal delivery. Traffic is not interrupted but takes some extra time.

Only safe rules are copied back from the Sandbox network to the operational network successfully protecting (S1) from being flooded with DOS attack rules.

The cost of the model is:

1. Double sized switch with extra ports of the same number of working ports connected to hosts.

2. Extra time is needed when forwarding the traffic to the sandbox switch and back.

Chapter 5: Results

Simulation Environment

The network was simulated using the mininet SDN simulator with its GUI (graphical user interface) is called MiniEdit. It's a python tool was used to build the network. Mininet implements a simulation environment for hosts and switches. Hosts are represented by Linux virtual terminals. Each host given an IP and can run basic Lunix processes like ping and hping. SDN switches are also simulated within mininet. The mininet comes with an internet controller, however, it supports OpenFlow for [R1] external controllers too [35].

We used Ryu OpenFlow as a controller to provide SDN environment by using Python scripts that support OpenFlow 1.3. Ryu is used to implement the sandbox controller model. The source code is modified of the Ryu controller as shown later on. Ryu is loaded and run on the same simulation machine [R2].

Ryu coveys between the Control Layer and the Infrastructure layer, for utilizing the OpenFlow convention. Likewise, the controller gives an API to create SDN applications that keep running in the Application Layer over the Control Layer. There are various OpenFlow controllers. The Ryu OpenFlow controller is one that utilizations Python scripts as its applications [36].

The mininet + Ryu system was executed as Linux processes on Ubuntu platform and Oracle virtual machine. We used Wireshark to analyse our result.

An Intel Xeon dual processor DELL TS 3500 workstation with 32 GB RAM and NVIDIA Quadro 2000 GPO was used to carry out the environments.

[R1] Mininet

[R2] Ryu

In the simulation, the same controller process was used to control both the working switch and the sandbox switch too, two separate controllers with a secure intercommunication channel should be used.

Acknowledgement of the CSE (Computer Science & Engineering) department that Ain Shams University cooperated in allowing me to conduct the experiments in their labs.

Experimental Setup

In mininet by default, it creates a basic network module, consists of one of each; switch, host, and controller. We just added another host, to be able to ping from host one to host two to trace route packets for detecting troubleshooting in the network. As mininet run on a trivial virtual machine that runs on the Linux kernel platform to resources only inside its namespace without taking much memory. Although, all the processes are operating on the computer and in the same memory space moreover, mininet shows an API to layout, links of a network between network features and required parameters for connection configuration, simultaneous bandwidth. We installed Mininet on a virtual machine instead of running it on a PC [36].

The topology contains less number of needed nodes to simulate the attacks and to analyses the results. We used one switch (s1) and two hosts (h1 and h2) which are linked. The switch is identified to the controller (c0) by a committed out of band control channel. We presume that the attacker runs one of the hosts (h1), which directly linked to OpenFlow switch. The attacker has no power over the controller, switch, or the control channel. The packet captures were taken with the hosts, the switch and the controller.

In the observations, the controller command the switch to utilize the OFPT FLOW MOD message to install a 7-tuple packet to emulate with the flow rule based on the information of packet header lines. The 7-tuple set contains of source MAC address, destination MAC address, source IP address, destination IP address, protocol, source port and destination port. The import or message inserts specification to the switch to help the packet to be sent to host h2.

Traffic is sent from source h1 to h2 that generated with hping3. Every packet has a source port, which operates a new regulation. In every trial, h1 sends a sum of 50000 UDP packets (with 512 bytes of payload) at the amount of about 1000 packets per second (˜4Mpbs).

Modified code of the Ryu Controller

A class with a rule base is added to the controller on the sand box

The algorithm is as follows:

1. When a new rule is added (due to an unknown packet) the rule is copied to the rule base using the function (rb_add_flow)

2. When a packet is removed due to a time out, the rule is removed from the rule base using the function (rb_remove_flow)

3. Periodically the rulebase is read and sent to operational controller, so it can call the add_flow function to add those rules.

When a new flow rule is added, this function is called

```
def add_flow(self, datapath, priority, match, actions, buffer_id=None):
    ofproto = datapath.ofproto
    parser = datapath.ofproto_parser
    inst = [parser.OFPInstructionActions(ofproto.OFPIT_APPLY_ACTIONS,
                                         actions)]
    if buffer_id:
        mod = parser.OFPFlowMod(datapath=datapath, buffer_id=buffer_id,
                                priority=priority, match=match,
                                instructions=inst)
    else:
        mod = parser.OFPFlowMod(datapath=datapath, priority=priority,
                                match=match, instructions=inst)
    datapath.send_msg(mod)
```

The method after Modification is to add a single function for registering this rule in the data base

```
rb_add_flow(datapath)
```

Receive Processing of FlowRemoved Message When LACP data units are not exchanged during the specified period, the OpenFlow switch sends a FlowRemoved message to the OpenFlow controller.

```
def flow_removed_handler(self, evt):
    """FlowRemoved event handler. when the removed flow entry was
    for LACP, set the status of the slave i/f to disabled, and
    send a event."""
    msg = evt.msg
    datapath = msg.datapath
    ofproto = datapath.ofproto
    dpid = datapath.id
    match = msg.match
    if ofproto.OFP_VERSION == ofproto_v1_0.OFP_VERSION:
        port = match.in_port
        dl_type = match.dl_type
    else:
        port = match['in_port']
        dl_type = match['eth_type']
    if ether.ETH_TYPE_SLOW != dl_type:
        return
    self.logger.info(
      "SW=%s PORT=%d LACP exchange timeout has occurred.",
      dpid_to_str(dpid), port)
    self._set_slave_enabled(dpid, port, False)
    self._set_slave_timeout(dpid, port, 0)
    self.send_event_to_observers(
      EventSlaveStateChanged(datapath, port, False))
```

The added code will be

```
rb_remove_flow(msg.datapath)
```

Added Functions

```
def rb_add_flow(datapath)
      RB_List.append(datapath)
def rb_remove_flow(datapath)
    RB_List.index(datapath)
      RB_List.remove(datapath)
def rb_update
      while len(RB_List) >0 :
            Nr=RB_List.pop()
            Send_To_WC(Nr)
```

Thread to Periodically check the rules

```
from apscheduler.scheduler import Scheduler

sched = Scheduler()

sched.start()

sched.add_interval_job(rb_update, seconds = N)

....

sched.shutdown()
```

Attack Results

The model is simulated on Mininet simulator [20] is used to test the proposal with the same attack described in section 4.

The total number of packet dropped is almost 0 meaning that the attack did not succeed.

One final comment is that it is possible to use the Mininet environment with software switches for implementing the Sandbox with real SDNs. A sufficient resourced computer will be able to mitigate DOS attacks with minimal cost.

Experimental Results:

The purpose of this experiment was to test the overall approach and provide an insight on the performance of the OpenFlow Sandbox solution.

Experiment Phases

The experiment was done in 2 phases: the first one was to show the attack on OpenFlow switches without the protection.

In this case, a simple 2 port switch was attacked with a DDOS attack using the HPING3 tool. The tool is used to generate traffic of UDP packets. In each packet, the destination port number is changed (incremented) causing the OpenFlow switch to add a new rule with each incoming packet flooding the flow tables in memory.

The HPING3 tool is a command line tool and the command used was:

```
hping3 -c 50000 -d 512 –udp -V -I u1000 10.0.0.2
```

Hping is an order line arranged TCP/IP parcels constructing analyzer that interface is stimulated to the ping, 8 unix charge however, ping isn't just ready to send ICMP imitation demands. Hping3 is not a packet initiation to expend for a scripting dialect but it is a scriptable security implementation [28].

This command is to generate 50,000 packets (-c switch) with a packet size of 512 bytes (-d switch). The packets are UDP packets. Each one is separated by 1milliseconds (1000 Micro seconds).

The attack is mounted from the first host (h1: 10.0.0.1) connected to port one and targeting the second host (h2: 10.0.0.2)

Wireshark was used to monitor the traffic at h2,

Phase one, No Sandbox

The results show that out of the 50,000 packets sent, only 45,418 packets were received. The drop ratio is about 10%.

This value was for the default timeout of the switch which was 10 seconds.

As shown in [16], the results vary with the time out settings and the figure below shows this variation

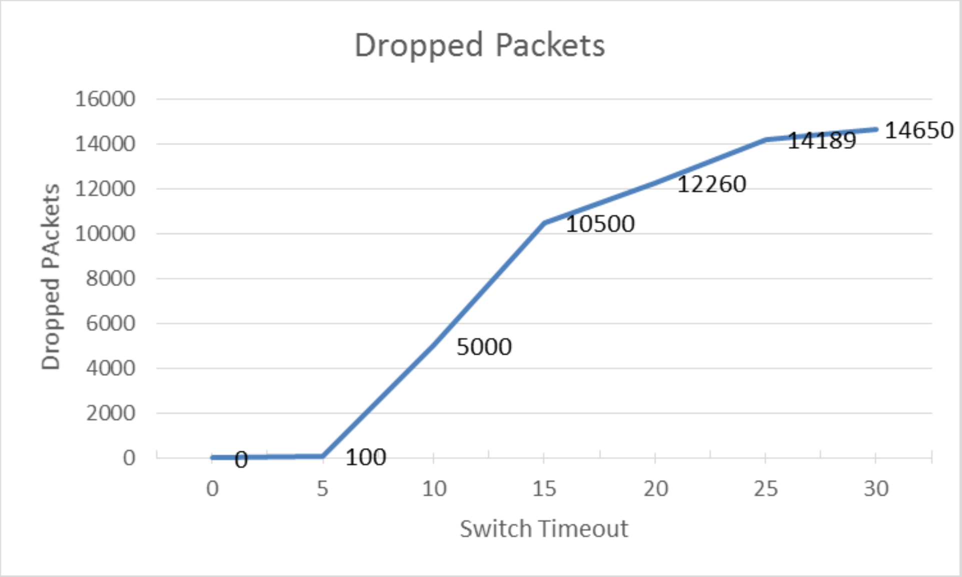


Figure 13. Phase one: No Sandbox.

The same setup was implemented again with the sandbox model in place.

The total numbers of dropped packets were 0 as a result of the sandboxing. The operational switch (S1) flow table was not flooded.

We utilize the traffic based OpenFlow switch execution, and it is introduced on an autonomous Linux have (i.e., programming switch), and we set the highest stream rules for this switch as 8,000, which is the same design for HP 5412zl switch. We utilize POX as the controller for more space, and it dispatches a basic layer 4 packets exchanging application along these lines, this application implements a stream manage with 4 tuples, granularity. Other two hosts are straightforward Linux, one is utilized for the proposed attack. And the other is utilized to run a TCP server program. For DOS attackers, attack packets to send numerous system parcels, whose 4-tuples are distinctive from each other. What's more, we control the packets per flow entry with 64 bits optional to be received with required 32 seconds duration time. As sending rate from 50 packets per second till most of the packet at least matched to dispatch different assault scenarios until process of actions of the pipeline stops executing packets [7].

```
        /* Group table commands */

        enum ofp_group_mod_command {

          OFPGC_ADD =0, /* Request new group entry. */

OFPGC_MODIFY = 1, /* Modify all matching groups added. */

OFPGC_DELETE = 2, /* Delete all invalid matching groups. */ };
```

Phase two, with Sandbox

In addition, if the attacker sends assault parcels, intensively. Attackers can devour all stream table entities in 2 or 3 seconds. It is a basic issue, on the grounds that even on the off chance that there may be a guarding way to deal with secure the data plane from the DOS attack on the off chance, that it doesn't quit sending assault packets in less seconds, and it can't secure the data plane [5].

Also, we measure the transmission capacity required for the proposed attack, and it is displayed in Figure 13. It demonstrates that the DOS attack requires around maximum transmission rate in the link, which is 200 Kbps (bytes every second) in most extreme and 20 Kbps in least. It infers that it is conceivable that an assailant can lead a generally stealthy DOS assault to a SDN classification. On the off chance that the aggressor can employ various has by leasing bot tainted hosts, he can direct a DOS assault with insignificant identification. For instance, if the assailant employs 100 has, each host just requires to send packets at the max-speed is 4 GB/s and current speed rounded 200 bps, which emulates an ordinary customer great and in this manner difficult to distinguish.

We can pack the stream administers by changing the condition of a stream rules to make the flows monitor more extensive ranges with arrange the request of an array of zero or may be more. For instance, if the goal port is a piece of a condition for a SDN organize giving burden adjusting capacity, we can change this part of the destination port to let a single stream control or handle many system streams, when there are as well many system streams to deal with. Obviously, it could demolish some heap adjusting strategies. In any case, it can make the SDN arrange still be working, and in some circumstance it is more vital to keep the network versatile from proposed DOS attack. Than safeguarding the stack adjusting strategy to the data plane.

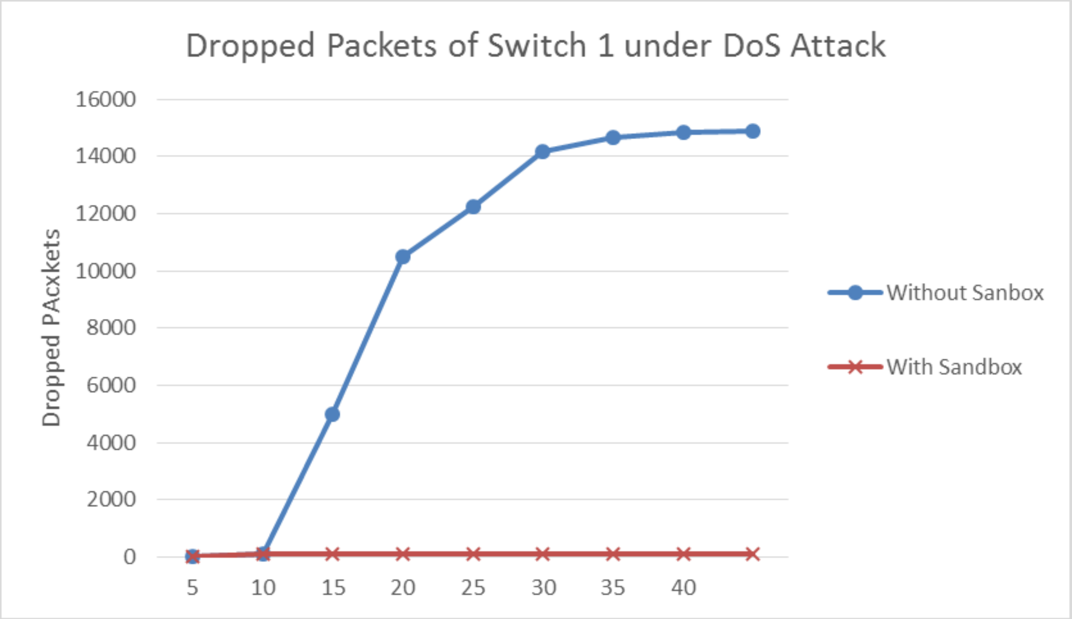


Figure 14. The DOS attack time and to set model bandwidth upon>0.

Through network performance of traffic analysis, this calculation manually done by Wireshark packet analyser that running on TCP protocols to evaluate the mechanism of metric by the presence flags of time intervals with value 0x0001 to eliminate implementation with minimum timeout, which is physically constant.

A constant Mbps rate application utilized UDP and was designed around 200 Kbps in 11,000 byte packets. Output drops are caused by the limitation of bandwidth and some traffic interface that had occurred on the running network. The traffic rate of the active interface is lower than the last experiment, figure 13. It can acknowledge almost all packets that ought to be conveyed upon traffic stability, if time is remain constant, which incremented in this bandwidth. In figure 14, output drops are caused by a congested interface in the network bandwidth that the movement rate of traffic interface till it can't acknowledge some packets to send out. A definitive answer solved the issue of high flow by expanding the line speed and using the sandbox, the blue line in figure 14 is the proof of our goal that we had more scalability in our theory. This approach eased the high rate flow to free flow capacity and controlled output drops packets upon the timeout. Also, output can be eliminated more in the future experiments, if output drops packets are an outcome of short bursts of information and the controllers can communicate easier with each other. On the off chance of maximum size of packets, output drops are still caused by a steady, high rate stream.

In performance, this network latency is indicated according to the time priority. The time required to handle a packet over a system. Inactivity might be measured in various courses: round trip, one way, and so forth. Inactivity might be affected by any component in fasten which is utilized information of workstation, connections, neighbourhood, switches, and server. At last it might be constrained, for extensive systems, by the speed of light. In spite of the largest level capacity of a table, the resource consumed by a single flow table entities will not consumed constant, depends on its match resources or parameters.

The TCP blockage window system manages missing acknowledgment packets by capture delay ACKs that impacted from the increased time as a side effect, if an acknowledgment packet is absent after a timeframe, the parcel is considered as lost and the TCP size window is diminished considerably as constant and the throughput as well that counting bytes of data sent, which relates to the impression of capacity limit on the route by the sender due to the time loss in retransmission. TCP window size would then be able to restart expanding if acknowledgment packets are gotten legitimately [33].

Packet misfortune will effects the speed of transmission of information: Parcels should be retransmitted. Regardless of the possibility that exclusive the acknowledgment packet got lost and the parcels got conveyed. The TCP clog window size won't permit an ideal throughput. At a constant time interval of delays data with less two percent of packet loss, the TCP throughput is in the vicinity should be lower than with no packet misfortune at OpenFlow hybrid switches.

Measuring Execution

These are suggested, however you're allowed to utilize any apparatus you're comfortable with.

Transmission capacity (bwm-ng, ethstats)

Latency (utilize ping for reachability)

Lines or queues (utilize tc incorporated into monitor.py)

TCP CWND insights (tcp_probe, perhaps we should add it to monitor.py)

CPU utilization (worldwide: top, or per container cpuacct)

Check OpenFlow header of each message from the controller

```
/* Header on all OpenFlow packets. */

struct ofp_header {

uint8_t version;   /* OFP_VERSION. */

uint8_t type;        /* one of the OFPT_ constants (OFPST_ => OFPMT_).*/

uint 16_t length;  /*Length including this ofp_header flags. */

uint32_t xid;          /*Transaction id associated with this packet. Replies use the same
id as was in the request to facilitate pairing.*/

};
```

The match types are structured utilizing OXM match classes. The OpenFlow designation distinguishes two types of OXM match classes, ONF member classes and ONF reserved classes, differentiated by their high order bit. Classes with the high order bit set to 1 are ONF reserved classes, they are utilized for the OpenFlow designation itself. Classes with the high order bit set to zero are ONF member classes, they are allocated by the ONF on an as needed substratum, and they uniquely identify an ONF member and can be used arbitrarily by that member. Support for ONF member classes is optional.

The following OXM classes are defined:

```
/* OXM Class IDs.
```

```
 * The high order bit differentiate reserved classes from member classes.

* Classes 0x0000 to 0x7FFF are member classes, allocated by ONF.

 * Classes 0x8000 to 0xFFFE are reserved classes, reserved for standardisation.

*/

enum ofp_oxm_class {

OFPXMC_NXM_0 = 0x0000, /* Backward compatibility with NXM
*/OFPXMC_NXM_1 = 0x0001, /* Backward compatibility with NXM */
OFPXMC_OPENFLOW_BASIC = 0x8000, /* Basic class for OpenFlow */
OFPXMC_EXPERIMENTER = 0xFFFF, /* Experimenter class */};
```

To defend SDN networks from the proposed attack, we now discuss some possible forfending methods. Firstly, we can compress the flow rules by transmuting the condition of a flow rule to make the flow rule cover wider ranges with wildcards. For example, if the destination port is a component of a condition for a SDN network providing load balancing function, we can transmute this component into the wildcard to let a single flow rule handle many network flows, when there are too many network flows to handle. Of course, it could ruin some load balancing policies. However, it can make the SDN network still be working, and in some situation it is more paramount to keep the network resilient from proposed DOS attack than preserving the load balancing policy [26].

Secondly, we can integrate some incipient functions that can detect this kind of scanning attack to the data plane. If a host creates some network flows that can create incipient flow rules in a short duration, we can consider it as suspicious. Then, we can ignore some network packets from this source for some time interval. However, this may not work for distributed assailing hosts like a botnet.

Thirdly, predicated on our detection results, we observe that if the standard deviation of the flow setup time is high may cannot efficaciously SDN network. We can utilize this characteristics to the proposed attack. For example, we can make the control plane varies the flow setup time dynamically. However, we should consider that the flow setup time should still meet some requisite that flow should enforced within 20ms in some cases.

Chapter 6: Conclusion and future work

This duplicator introduced some basic description of SDNs and OpenFlow protocol and components. The possibility of attacking OpenFlow switches utilizing the openness of OpenFlow causing the flow tables of the switch to be flooded is described.

A unique novelty of the Sandbox with SDN, protection model was introduced, and proven to mitigate the DOS attack which has been presented successfully.

The Mininet environment was used for simulation and also was proposed to implement the actual sandbox for minimizing the cost. Only the rules time-out was used to judge the validity of the rule and only one attack model is addressed.

In the future, it is recommended to consider more attacks, especially sophisticated ones and also to implement more powerful analysis tools on the sandbox controller so as to detect more attacks. This may include information from all over the network devices including security systems and operating system logs, and may also consider the payload of the packets.

Bibliography

[1] Jakob Spooner, Dr Shao Ying Zhu, “A Review of Solutions for SDN-Exclusive Security Issues,” *(IJACSA) International Journal of Advanced Computer Science and Applications,* vol. 7, 2016.

[2] S. Shin and G. Gu, “Attacking Software-defined Networks: A First Feasibility Study, in Proceedings of the Second ACM SIGCOMM,” *Workshop on Hot Topics in Software Defined Networking, ser. HotSDN'13, New York, NY, USA: ACM,* pp. 165-166, 2013.

[3] Haopei Wang, Lei Xu, Guofei Gu, “FloodGuard: A DoS Attack Prevention Extension in".Software-Defined Networks.”.

[4] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker and J. Turner, “OpenFlow: enabling innovation in campous networks,” *ACM SIGCOMM Computer Communication Review,* vol. 38, pp. 69-74, 2008.

[5] Khattab, S., Sangpachatanaruk, C., Znati, T., Melhern, R., and Mosse, D. (2003), “ Proactive server roaming for mitigating denial-of-service attacks.,” *Proceedings of Conference on Information Technology Research and Education (ITRE’03),* p. 1–5.

[6] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka and T. Turletti, “A survey of software-defined networking: Past, present, and future of

programmable networks," *IEEE Communications Surveys and Tutorials,* vol. 16, no. 3, pp. 1617-163, 2014.

[7] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow.," in *In Local Computer Networks (LCN), IEEE 35th* , October 2010.

[8] A. M. Bahaa-Eldin, "TARA: Trusted Ant Colony Multi Agent Based Routing Algorithm for Mobile Ad-Hoc Networks," *in Bio-inspiring Cyber Security and Cloud Services: Trends and Innovations, Springer Berlin Heidelberg,* pp. 151-184, 2014.

[9] Aissani, A. (2008) , "Queueing analysis for networks under DoS attack.," *Proceedings of international conference on Computational Science and its Applications (ICCSA'08), Part II, Perugia, Italy,* 30 June - 3 July.

[10] Sung, M. and Xu, J. (2002) , "IP traceback-based intelligent packet filtering: A novel technique for defending against internet DDoS attacks.," *Proceedings of International Conference on Network Protocols (ICNP'02), Paris, France. IEEE Computer Society, Washington, DC, USA.,* p. 302–311.

[11] S. Mousavi and M. St-Hilaire, "Early Detection of DDoS Attacks against SDN Controllers," *International Conference on Computing Networking and Communications,* pp. 77-81, 2015.

[12] A. M. Bahaa-Eldin, "A Bio-inspired Comprehensive Distributed Correlation Approach for Intrusion Detection Alerts and Events," in Bio-inspiring Cyber

Security and Cloud Services: Trends and Innovations," *Springer Berlin Heidelberg,* pp. 3-38, 2014.

[13] O. Abdelrahim, A. Taha and A. M. Bahaa-Eldin, "A framework for virtual machine admission control in cloud environment," *The 11th IEEE International Conference on Computer Engineering and Systems,* 2016.

[14] M. Mousa, M. Sobh and A. Bahaa-Eldin, , "Software Defined Networking Concepts and Challenges," *The 11th IEEE International Conference on Computer Engineering and Systems, ,* 2016.

[15] Ruslan Kirichek, Andrei Vladyko, Maxim Zakharov, Andrey Koucheryavy, "Model Networks for Internet of Things and SDN," *Department of Telecommunication Networks and Data Transmission, St.Petersburg State University of Telecommunication,* 2016.

[16] H. N. Gabra, A. M. Bahaa-Eldin and H. K. Mohammed, "Data Mining Based Technique for Ids Alert Classification," *International Journal of Electronic Commerce Studies,* vol. 6, no. 1, pp. 119-126, 2015.

[17] Mousa, Mohammad, Ayman M. Bahaa-Eldin, and Mohamed Sobh, "Software Defined Networking concepts and challenges In Computer Engineering & Systems (ICCES)," *IEEE 11th International Conference* , pp. 79-90, 2016 .

[18] A. E. Taha, I. Abdel Ghaffar, A. M. Bahaa-Eldin and H. M. Mahdi, "Agent based correlation model for intrusion detection alerts," *IEEE International Conference on Intelligence and Security Informatics (ISI),* 2010.

[19] A. M. Bahaa-Eldin, “Time series analysis based models for network abnormal traffic detection,” *International Conference on Computer Engineering and Systems (ICCES), IEEE,* 2011.

[20] H. M. A. F. a. A. M. B.-E. I. T. Abdel-Halim, “Agent-based trusted on-demand routing protocol for mobile ad-hoc networks,” *Wireless Networks*, vol. 21, no. 2, pp. 467-483, 2015.

[21] A. O. A. El-Mal, M. A. Sobh and A. M. Bahaa-Eldin, “Hard-Detours: A new technique for dynamic code analysis,” *IEEE EUROCON,* 2013.

[22] D. Plummer, “Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48. bit Ethernet address for transmission on Ethernet hardware,” *IETF, Internet Standard RFC:826,* 1982.

[23] R. Kandoi and M. Antikainen, “Denial-of-service attacks in OpenFlow SDN networks,” *IFIP/IEEE International Symposium on Integrated Network Management (IM),* pp. 1322-1326, 2015.

[24] Q. Duan, E. Al-Shaer and H. Jafarian, “Efficient Random Route Mutation considering flow and network constraints,” *Conference on Communications and Network Security (CNS),* pp. 260-268, 2013.

[25] December 1st 2016. [Online]. Available: http://cpqd.github.io/ofsoftswitch13/.

[26] Shirali-Shahreza, Sajad and Y. Ganjali FleXam, “ Flexible sampling extension for monitoring and security applications in openflow flexible sampling

extension for monitoring and security applications in openflow," *SIGCOMM workshop on Hot topics in software defined networking,* pp. 167-168, 2013.

[27] Seungwon Shin, Vinod Yegneswaran, Phillip Porras, Guofei Gu, " AVANT-GUARD: Scalable and Vigilant Switch Flow Management in Software-Defined Networks," p. 4–8 , November 2013.

[28] J. Mirkovic, and P. Reihe, "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms.," *ACM SIGCOMM Computer Communications Review,* vol. 34, April 2004.

[29] R. Kl¨oti, V. Kotronis and P. Smith, "OpenFlow: A security analysis," *21st IEEE International Conference on Network Protocols (ICNP),* pp. 1-6, 2013.

[30] M. A. Mashrei, "Neural Network and Adaptive Neuro-Fuzzy Inference System Applied to Civil Engineering Problems," *Thi-Qar University, College of Engineering, Civil Department,* 2012.

[31] Srinivas Mukkamala, Guadalupe Janoski, Andrew Sung , "Intrusion Detection: Support Vector Machines and Neural Networks," *New Mexico Institute of Mining and Technology* .

[32] Krzysztof Cetnarowicz, Renata Ci¦ciwa, Gabriel Rojek, "Behavior Evaluation with Earlier Results Collectionin Multi Agent System," *Inteligencia Artifcial* , vol. 9, 2005.

[33] Georgios Loukas and Gulay Oke, “Protection against Denial of Service Attacks: A Survey,” *The Computer Journal, Intelligent Systems and Networks Group, Imperial College London, Oxford University,* 6 May 2009.

[34] Ningning Lu, Huachun Zhou, Hongke Zhang, “A NEW PROBABILISTIC PACKET MARKING TECHNOLOGY BASED ON PATH IDENTIFICATION,” 2009.

[35] Jacek Chrzaszcz, Tomasz Stachowicz, Andrzej Gasienica-Samek, and Aleksy Schubert, “Minik: A Tool for Maintaining Proper Java Code Structure,” *IFIP International Federation for Information Processing,* vol. 227, pp. 361-371, 2006.

[36] Uttam Ghosh, Xinshu Dong, Rui Tan, Zbigniew Kalbarczyk, David K. Y. Yau, Ravishankar K. Iyer, “A Simulation Study on Smart Grid Resilience under Software-Defined Networking Controller Failures,” 2016.

Appendix: Steps of the implementation

Preparation

A. In your VMware, run the VM and make sure the necessary programs and tools are exist there.

B. For instance: If hping3 is not there, it should be installed. Open a terminal and write:

1. sudo apt-get update
2. sudo apt-get install hping3

C. In python language, create a text file and write a simple topology of your network (h1, h2, s1, s2, and c0) then save it in the directory “Mininet/Custom”.

D. In python language, create a text files about the Sandbox switch and a simple switch in “Ryu/App” directory.

Create DoS Attack Firstly

A. Open a terminal then write:

sudo mn --custom ~/mininet/custom/topo-2sw-2host.py --topo mytopo --mac --switch ovsk --controller remote –x

Five screens will be opened, they are titles h1, h2, s1, s2, and c0

B. **In s1 screen:** ovs-vsctl set Bridge s1 protocol=OpenFlow13

C. **In s2 screen:** ovs-vsctl set Bridge s2 protocol=OpenFlow13

D. Open a new terminal and then write:

cd ryu/app

sudo gedit “simple switch folder name”.py

E. Scroll in that file until you find the lines

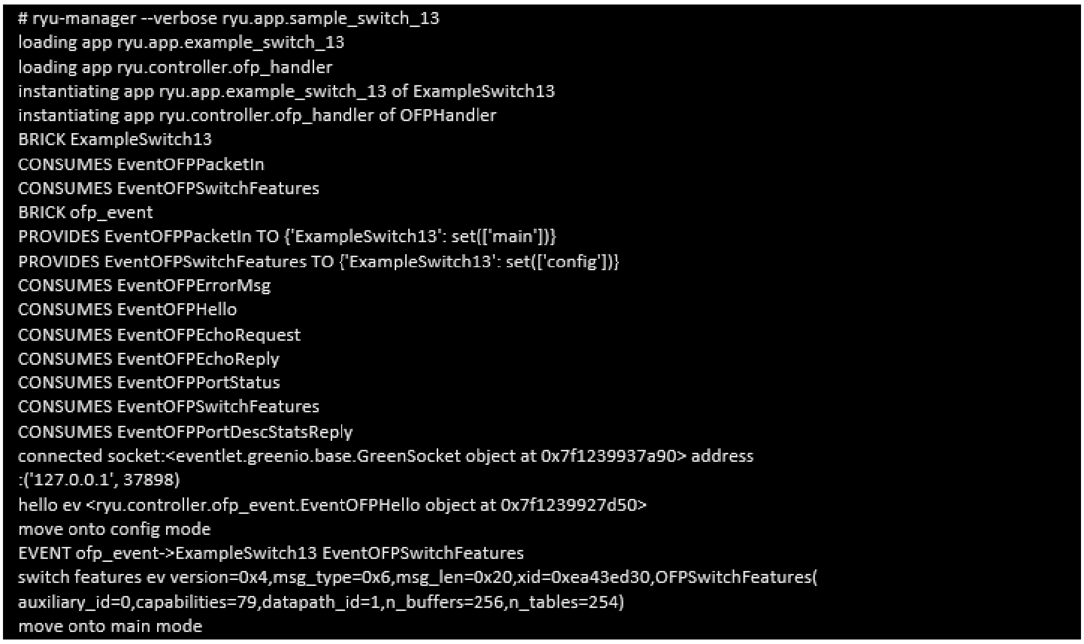
Global Rules Timeout

Rules Timeout=100

Change the values of this filed (Rules Timeout) for any value from 0 to 100 and "Save". Start with 100, and make sure to not enter a value above 100.

F. **In c0 screen:** ryu-manager --verbose ryu.app.simple_switch_13

Wait until you see something like this and Note the last line



```
# ryu-manager --verbose ryu.app.sample_switch_13
loading app ryu.app.example_switch_13
loading app ryu.controller.ofp_handler
instantiating app ryu.app.example_switch_13 of ExampleSwitch13
instantiating app ryu.controller.ofp_handler of OFPHandler
BRICK ExampleSwitch13
CONSUMES EventOFPPacketIn
CONSUMES EventOFPSwitchFeatures
BRICK ofp_event
PROVIDES EventOFPPacketIn TO {'ExampleSwitch13': set(['main'])}
PROVIDES EventOFPSwitchFeatures TO {'ExampleSwitch13': set(['config'])}
CONSUMES EventOFPErrorMsg
CONSUMES EventOFPHello
CONSUMES EventOFPEchoRequest
CONSUMES EventOFPEchoReply
CONSUMES EventOFPPortStatus
CONSUMES EventOFPSwitchFeatures
CONSUMES EventOFPPortDescStatsReply
connected socket:<eventlet.greenio.base.GreenSocket object at 0x7f1239937a90> address
:('127.0.0.1', 37898)
hello ev <ryu.controller.ofp_event.EventOFPHello object at 0x7f1239927d50>
move onto config mode
EVENT ofp_event->ExampleSwitch13 EventOFPSwitchFeatures
switch features ev version=0x4,msg_type=0x6,msg_len=0x20,xid=0xea43ed30,OFPSwitchFeatures(
auxiliary_id=0,capabilities=79,datapath_id=1,n_buffers=256,n_tables=254)
move onto main mode
```

Figure 15. Output of c0 screen.

G. **In h2 terminal start tcpdump:** tcpdump -XX -n -i h2-eth0 -w ~/mininet/ “Enter a name to this file”.pcap

H. **In h1 screen write:** hping3 -c 50000 -d 512 --udp -V -i u1000 10.0.0.2 (Do this immediately without delays) from step F.

* Note the maximum seq=xxx in the h1 screen, this is the number of packets send and then the switch will hang, if the rules Timeout=100 then the seq=3000 almost should be seen, this means 47000 packet is lost (50000-3000)

I.

A. Press CTL+C in both h1 and c0 screens.

B. Go to step E and decrease the number from 100 to 50 and then "Save".

C. Repeat the steps of F and H, note the seq=xxx increases say to ~30000.

Finally, repeat all the above for other values of Rules Timeout until you reach 0, In case of 0, the hping3 will continue until its end, this means no packet lost and DOS is not possible but note the c0 screen will print a lot of messages.

Second: Sandbox

J. Now for the sand box, go to the Mininet terminal and in the Mininet> console write:

Exit (mininet>exit)

K. Repeat the previous steps again but in all steps replace "simple switch" to "sand switch”.

L. **Start Wireshark in a new terminal:** sudo wireshark &

M. Open the dump file and check the statistics.