DDoS Mitigation In SDN

[1]Navdeep Singh, [2]Supreet Kaur
[1]Student, [2]Assistant Professor
[1]Dept. of Computer Science and Engineering,
[1]Punjabi University, Patiala, India

Abstract - SDN is changing the networks around the world by changing the traditional way of control plane and data plane working. All the network with SDN is controlled using SDN controller and due to its centralized management, it brings large number of benefits to network industry. There are some limitations also which are under review in this paper like Security of the SDN controller. As centralization of the network means that it can be controlled from a single device and all the control plane is implemented using the single controller and its security is vital in the SDN. DDOS attacks can create a lot of issues in the network and can disrupt the whole data center network services integrated with the controller. This paper explains the DDOS attacks and how they can be implemented using hping tool in Kali Linux to disrupt the services of the controller and how it can be prevented by using the inbuilt Linux Firewall.

Keywords- Software Defined Networking, Openflow, Control Plane, Data Plane, DDoS, iptables, hping

I. INTRODUCTION

Software Defined Networks is decoupling of control plane with data plane in the network. It is the result of Clean Slate Project by Martin Cassado, where Martin wants to research to achieve the objective i.e. What if Internet is created from scratch, How different and better it can be made?. SDN changed the whole network scenario of traditional networks where every device has its own control plane and data plane, SDN brings the centralization in the network by introducing the single point of control plane known as Controller from where all other devices which acts as Data Plane can be managed and path selection is depended on that Controller. For redundancy purposes, multiple controllers can also be used in Clustered Manner. Redundant SDN Controller helps in more reliability as there is only single point of failure in network, therefore a backup always is a better option. SDN architecture is also quite different from the traditional network. Traditional Network devices like routers and switches controls their data plane locally and each devices has to be configured with protocols to make the control plane work. Also proprietary devices before only provides functionality which is offered by the vendor and those devices and the network operating systems related with vendors lack flexibility or customization. Below is the figure displaying the comparison between traditional networking and software defined networking:

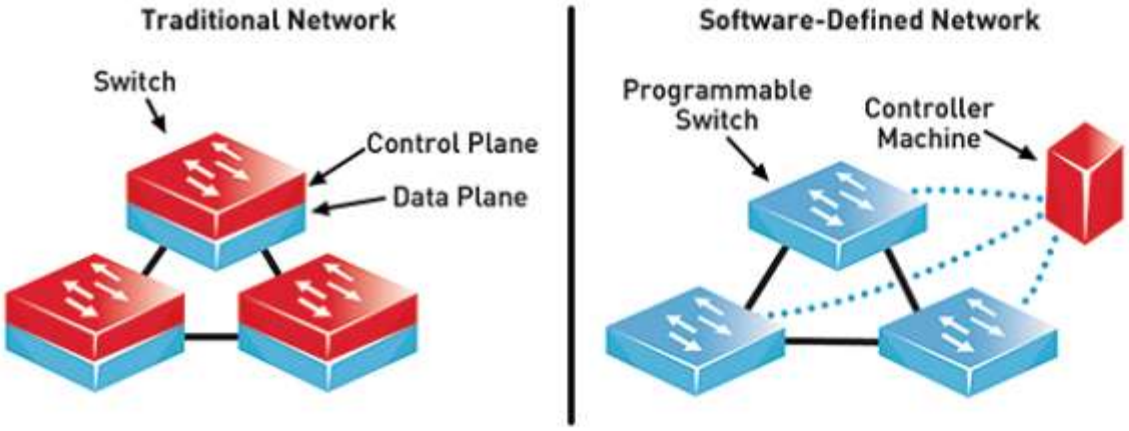


Fig. 1.1 Traditional Network vs SDN

II.SDN BENEFITS

- **Enhancing Configuration** - SDN helps in reducing this problem, the best thing about SDN is that all the configuration of network devices can be done using a single controller that takes less amount of time and helps in faster troubleshooting.
- **Faster Innovation** - SDN provides an programmable network platform where new ideas can be deployed easily. As with the changing networks, it is always better to innovate in fast manner.
- **Lesser Network Infrastructure Costs -** One of the biggest benefits is lesser Network Infrastructure costs as there is no need to buy proprietary routers and switches and instead of using every device with control plane and data plane features.
- **Granular Security -** As more and more devices connecting with the Internet with IoT and cloud computing, information security is creating new challenges where SDN can work as single point of control for information security policies.
- **Better Visibility into the Network -** Another great benefit of using SDN is that with SDN, visibility into an organization's network become far better than traditional networks.

- **Better Uptime and more reliable network -** SDN supports the power of virtualization to the network and network devices can be replaced or upgraded without taking the system offline. In case any downtime happens, one can also make snapshots of configuration which can also helps in recovering outages at a rapid pace.

III.SDN CHALLENGES

- Tackling Fast On-Demand Growth
- Addressing automatic real-time changes
- Security

SDN Security - This is something which is very important part of SDN and a very big challenge also as protection of controller is very important in SDN because in case controller is compromised, the whole will also be compromised. New Applications when added can also introduce some security threats may be because of some bugs in the code of the applications, which leads to security related issues. There are lots of threats that SDN controllers face like Distributed Denial of Service-DDoS, Man-in-the-Middle-MiTM, Spoofing etc.

DDoS attacks can be used by hackers to disrupt the availability of the network by attacking the controller. Huge amount of traffic can be sent to controller by using botnet which makes the controller unavailable. Attacks can be done using different DDoS methods like TCP SYN Flood, ICMP Flood, UDP Flood etc.

IV.SDN CONTROLLERS

SDN revolves around the controller which acts as the intermediate layer between the application and infrastructure layer. SDN Controller controls the control plane of the network domain and provide instructions to the data plane based bare-metal switches on selection of best path towards destination and various other policies regarding security and QoS can be controlled using controller. Various SDN controllers which are used worldwide are Shown in table 1 :-

Controller	**Company**	**Open Source**
NOX	Nicira	Yes
POX	Nicira	Yes
Beacon	Stanford University	Yes
Maestro	Rice University	Yes
Floodlight	Big Switch Networks	Yes
Floodlight-Plus	Big Switch Networks	Yes
Ryu	NTT Labs	Yes
Open Daylight	Linux Foundation	Yes
Open vSwitch	Open vSwitch	Yes
Open Contrail	Open Contrail	Yes

Table – SDN Controller

V.DDoS Attacks

DDoS attacks are difficult to detect as they may seem as genuine packets and also as they are coming via botnet, therefore it's very difficult to detect the attacker. Due to these difficulties in detection of the attacker and attack, it is mostly used by attackers to disrupt the network infrastructures of the organizations. DDoS can also be used as a distraction by the attackers while they have a different motive, they may want companies to get confused that it is a DDoS attack to disrupt the network resources, but the target is tricked by hackers to steal the data from their data centers. A very prime example of this sort of attack is when a small ISP from London named "Talk Talk" is attacked by the hackers. Hackers performed a DDoS[16] attack on Talk Talk and suddenly their web services are disrupted and the accessibility of their website is disrupted. All the Talk Talk network team went on to troubleshooting the DDoS attack and put their all focus on resolving that issue while attackers get entry through the backdoor link and steals the customer records from Talk Talk Data Center.

DDoS attacks are increasing every year with attacks over 50Gbps are detected by various service providers. Due to emergence of Cloud and IoT, botnet is becoming more powerful and hackers get the bigger playground to play with. Mirai Botnet, which is one of the largest botnet ever, was created with the help of routers and Security Cameras. Weak security on these devices makes them easy pickings for the hackers. Average IoT devices are attacked every two minutes by the hackers. Some of the major targets of Mirai Botnet was cloud related services like DNS provider Dyn. This, along with millions of MongoDB databases which were hosted in the cloud. According to Symantec, average organizations use around 900 cloud based applications, while their CIOs think they are using around 30, which leads to large scale of inconsistency and underestimated level of risk. Attack on Dyn[16][18], also affects large scale companies like Paypal, Netflix and Spotify**.**

VI.DDoS Attack Types

- Performance of different traffic types over MPLS ***Application Layer DDoS attack:*** These are the attacks[19] which target Softwares like Apache, Windows IIS, or other software vulnerabilities to generate an attack and disrupt the services.
- ***Protocol DDoS attack:*** These types of attacks[11] are made at the protocol level. It includes TCP SYN Flood, Ping of Death etc.
- ***Volume Based DDoS attack:*** This type of attack uses ICMP Floods, UDP Floods, etc via spoofed packets.

Some of the most commonly used DDoS attack types include:

1) ***UDP Flood:*** It is a DDoS attack that targets with the help of UDP packet flooding. It attacks by flooding on random pots on a remote host. This makes host repeatedly checks for the application running on that ports and then reply with the Destination Unreachable message. This type of attacks can be made on applications using UDP like Video Conferencing Services and conference application can be disrupted using UDP Flood attack[5].

2) ***ICMP (Ping) Flood:*** It is pretty much similar to UDP Flood attacks, it creates a ICMP Request flood and send it to the target machine. This type of attack mainly utilize both outgoing and incoming bandwidth, therefore they are mainly performed using botnets. This type of attack can work on almost all the applications to choke the bandwidth of the network resources[5].

3) ***SYN Flood:*** In TCP SYN Flood attack, attacker sends SYN Floods[5][6] to the target machine and target machine replies back with the SYN-ACK and needs a TCP ACK in return from the attacking machine, but attacker never sends that back which results in target machine stucks in waiting state for all the TCP 3-Way Handshakes. It can be used to choke down applications running TCP based applications. Different types of tools can be used to attack TCP SYN Flood, which can create bots and then a botnet before starting an attack like Hping which comes packaged with the penetration testing distribution of linux like Kali Linux. Figure below displays the normal 3 way handshake against the SYN Flood attack where the 3 way handshake never completes.with Plain Traffic with no QoS and with QoS applied on it. By default when traffic enters into Service Provider Core Network from Customer, MPLS labels are attached to the IP packets and all the forwarding is done on the basis of the experimental bits. Below is the topology used for MPLS QOS and TE :-

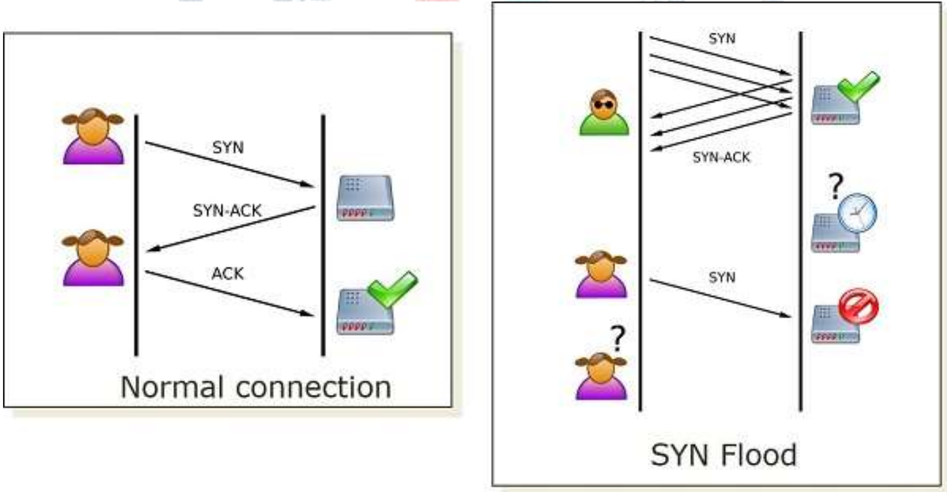


Figure – TCP SYN Flood vs Normal connection.

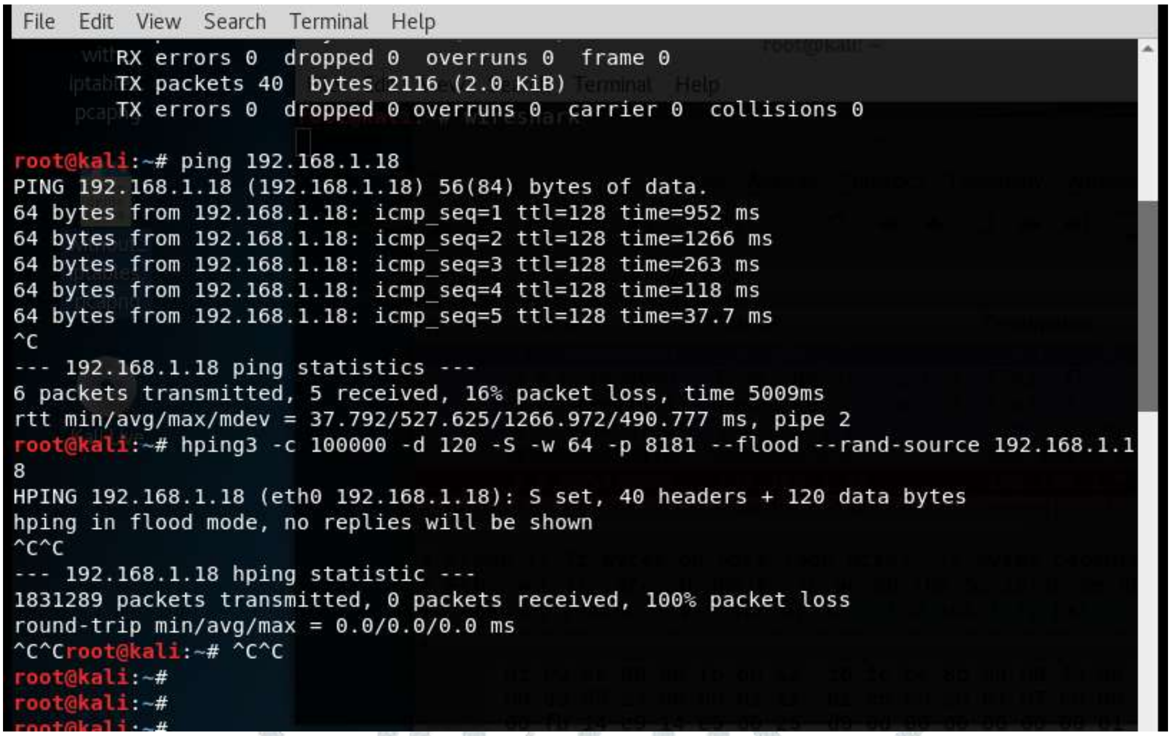
4) ***Ping of Death:*** In Ping-of-Death attack, attacker sends flood of malicious pings to the target machine. Attacker sends maximum sized IP packets to the target and it is split over multiple fragments and on the target end, he has to reassemble all the packets and when the target ends up with reassembling of packets, it ends up with packet size larger than 65535 bytes(maximum packet size) and slowly overflow memory buffers are allocated to the packet resulting in DoS attack[3].

5) ***HTTP Flood:*** In HTTP Flood attack, hacker exploits the target by sending HTTP GET or POST requests to web server or web application. This attack requires less amount of bandwidth than other attacks. Target Machine can be choked by sending hundreds of requests or by sending lots of Post messages which can disrupt the services of the web application or web site by choking the bandwidth[3].

VII.RESULTS

Hackers can disrupt SDN Controller Services and then ultimately take down the whole data center networks which are integrated with the SDN Controller. We have used OpenDaylight Controller in our testing and used Kali Linux for DDOS attack. In kali linux,we have used HPing tool to attack SDN controller. Both ODL and Kali Linux are running in two Virtual Machines and having following IP addresses:

- Kali Linux – 192.168.1.180
- ODL – 192.168.1.18
- Below is the screenshot showing DDOS attack on ODL using Hping command line tool:



```
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 40  bytes 2116 (2.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@kali:~# ping 192.168.1.18
PING 192.168.1.18 (192.168.1.18) 56(84) bytes of data.
64 bytes from 192.168.1.18: icmp_seq=1 ttl=128 time=952 ms
64 bytes from 192.168.1.18: icmp_seq=2 ttl=128 time=1266 ms
64 bytes from 192.168.1.18: icmp_seq=3 ttl=128 time=263 ms
64 bytes from 192.168.1.18: icmp_seq=4 ttl=128 time=118 ms
64 bytes from 192.168.1.18: icmp_seq=5 ttl=128 time=37.7 ms
^C
--- 192.168.1.18 ping statistics ---
6 packets transmitted, 5 received, 16% packet loss, time 5009ms
rtt min/avg/max/mdev = 37.792/527.625/1266.972/490.777 ms, pipe 2
root@kali:~# hping3 -c 100000 -d 120 -S -w 64 -p 8181 --flood --rand-source 192.168.1.1
8
HPING 192.168.1.18 (eth0 192.168.1.18): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
^C^C
--- 192.168.1.18 hping statistic ---
1831289 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
^C^Croot@kali:~# ^C^C
root@kali:~#
root@kali:~#
root@kali:~#
```

Figure 7.1 - hping command for performing ddos

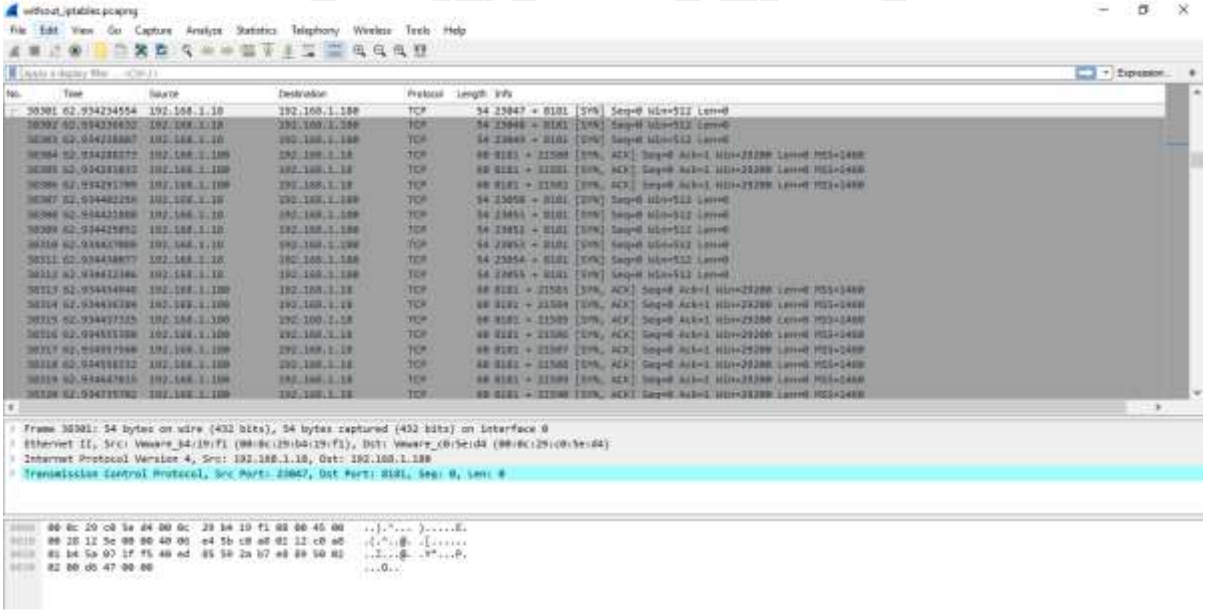
In the last attack using hping, we have used the command and have added some attributes which are explained below:

- **-c - --*count count***

Stop after sending (and receiving) *count* response packets.

- -d – data size
- -S – TCP SYN Flag set
- -w – tcp window size
- -p – port number
- --flood – hping will flood packets.
- --rand-source – hping will send packets with random source addresses.

After around 30 seconds of attack, SDN controller goes down and one of the reason behind that is the TCP SYN Flood that hping has created and controller replies every packet with a SYN-ACK and last attacker won't reply with last ACK to complete the three way handshake to make the sessions stuck in waiting state. Below is the wireshark capture showing the TCP SYN Flood attack with ODL responds with SYN-ACK:



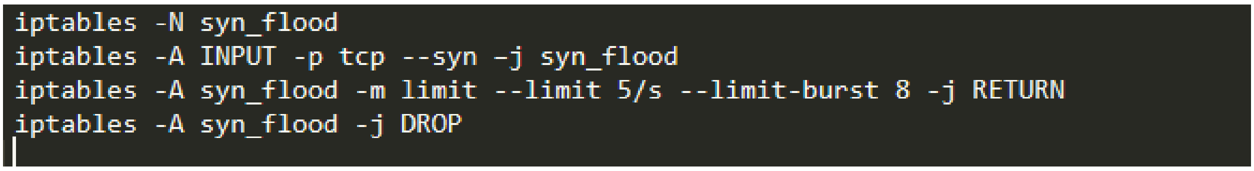
```
No.   Time          Source         Destination     Protocol Length Info
38301 62.934234554  192.168.1.10   192.168.1.100   TCP      54 23047 → 8181 [SYN] Seq=0 Win=512 Len=0
38302 62.934236432  192.168.1.10   192.168.1.100   TCP      54 23048 → 8181 [SYN] Seq=0 Win=512 Len=0
38303 62.934238887  192.168.1.10   192.168.1.100   TCP      54 23049 → 8181 [SYN] Seq=0 Win=512 Len=0
38304 62.934288272  192.168.1.100  192.168.1.10    TCP      60 8181 → 21580 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
38305 62.934281832  192.168.1.100  192.168.1.10    TCP      60 8181 → 21581 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
38306 62.934291799  192.168.1.100  192.168.1.10    TCP      60 8181 → 21582 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
38307 62.934402250  192.168.1.10   192.168.1.100   TCP      54 23050 → 8181 [SYN] Seq=0 Win=512 Len=0
38308 62.934423888  192.168.1.10   192.168.1.100   TCP      54 23051 → 8181 [SYN] Seq=0 Win=512 Len=0
38309 62.934425852  192.168.1.10   192.168.1.100   TCP      54 23052 → 8181 [SYN] Seq=0 Win=512 Len=0
38310 62.934437888  192.168.1.10   192.168.1.100   TCP      54 23053 → 8181 [SYN] Seq=0 Win=512 Len=0
38311 62.934438877  192.168.1.10   192.168.1.100   TCP      54 23054 → 8181 [SYN] Seq=0 Win=512 Len=0
38312 62.934412186  192.168.1.10   192.168.1.100   TCP      54 23055 → 8181 [SYN] Seq=0 Win=512 Len=0
38313 62.934434940  192.168.1.100  192.168.1.10    TCP      60 8181 → 21585 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
38314 62.934436284  192.168.1.100  192.168.1.10    TCP      60 8181 → 21584 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
38315 62.934437325  192.168.1.100  192.168.1.10    TCP      60 8181 → 21589 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
38316 62.934555388  192.168.1.100  192.168.1.10    TCP      60 8181 → 21586 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
38317 62.934557940  192.168.1.100  192.168.1.10    TCP      60 8181 → 21587 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
38318 62.934558232  192.168.1.100  192.168.1.10    TCP      60 8181 → 21588 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
38319 62.934447815  192.168.1.100  192.168.1.10    TCP      60 8181 → 21589 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
38320 62.934735782  192.168.1.100  192.168.1.10    TCP      60 8181 → 21590 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460

> Frame 38301: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
> Ethernet II, Src: Vmware_b4:19:f1 (00:0c:29:b4:19:f1), Dst: Vmware_c0:5e:d4 (00:0c:29:c0:5e:d4)
> Internet Protocol Version 4, Src: 192.168.1.10, Dst: 192.168.1.100
> Transmission Control Protocol, Src Port: 23047, Dst Port: 8181, Seq: 0, Len: 0

0000  00 0c 29 c0 5e d4 00 0c  29 b4 19 f1 08 00 45 00   ..).^... ).....E.
0010  00 28 12 5e 00 00 40 06  e4 5b c0 a8 01 12 c0 a8   .(.^..@. .[......
0020  01 b4 5a 07 1f f5 40 ed  85 58 2a b7 e8 89 50 02   ..Z...@. .X*...P.
0030  02 00 d6 47 00 00                                  ...G..
```

Figure 7.2 – Wireshark capture showing SYN-Flood on ODL

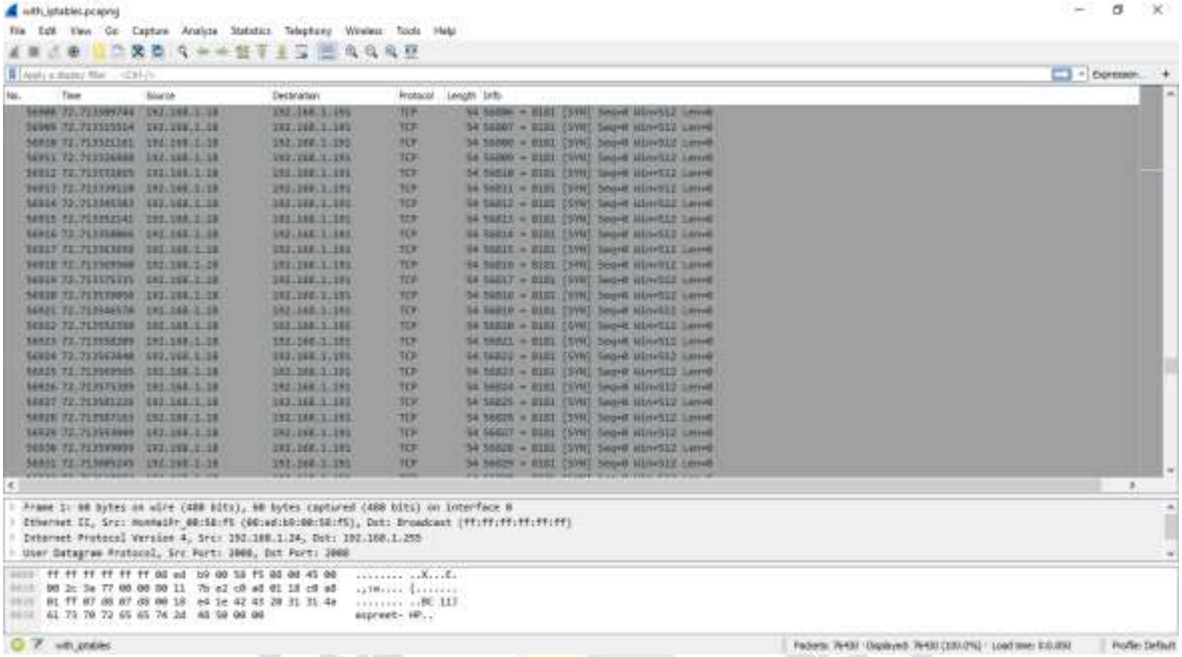
One of the reasons behind this attack being successful is that there is no DDOS protection at victim server machine or at their ODL Network. As ODL or almost all the SDN controllers run over Linux, we can use iptables, which is a linux based firewall, which helps us in reducing the impact the DDOS attack if we create a rule which limits the amount of TCP SYN-ACK responses per second according to the peak utilization of requests that ODL has before. To say, ODL has max 5 new TCP connections every second in its peak load. Therefore in Linux based firewall, i.e. iptables, we can create a policy script for that:



```
iptables -N syn_flood
iptables -A INPUT -p tcp --syn -j syn_flood
iptables -A syn_flood -m limit --limit 5/s --limit-burst 8 -j RETURN
iptables -A syn_flood -j DROP
```

Figure 7.3 – IPTables policy to mitigate DDOS

We have deployed the script on another ODL VM and it has reduced the impact on ODL machine as the limit is reduced to 5 TCP new session responses per second. After deploying the iptables script, there are no SYN-ACK reponses to the hacker more than 5, below wireshark capture screenshot shows the attacker is sending TCP SYN requests and is not getting SYN-ACKS back from the controller because of the iptables script that we have used.



with_iptables.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
No. Time Source Destination Protocol Length Info
with_iptables

```
Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
Ethernet II, Src: HonHaiPr_00:58:f5 (00:ad:b9:00:58:f5), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 192.168.1.24, Dst: 192.168.1.255
User Datagram Protocol, Src Port: 2008, Dst Port: 2008

ff ff ff ff ff ff 00 ad  b9 00 58 f5 08 00 45 00   ........ ..X...E.
00 2c 3e 77 00 00 80 11  7b e2 c0 a8 01 18 c0 a8   .,>w.... {.......
01 ff 07 d8 07 d8 00 18  e4 1e 42 43 20 31 31 4e   ........ ..BC 11N
41 73 70 72 65 65 74 2d  48 50 00 00               Aspreet- HP..
```

Figure7.4: Wireshark capture of TCP SYN from attacker after iptables script

As attacker’s TCP SYN requests are not getting any response and controller is continuously dropping TCP SYN Packets without replying, therefore attack will not get successful. Below are the bar graphs showing the output in more precise manner. Figure below shows the amount of packets we have sent in around an minute to perform a DDOS attack on controller with and without security mechanism:

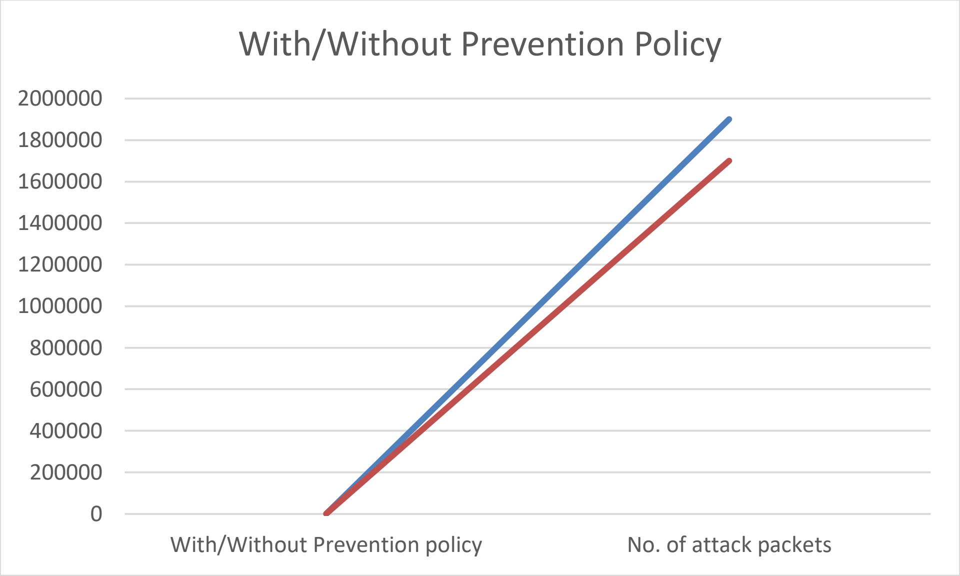


Figure7.5 :No of attack packets sent in around two minute on controller machines with or without security.

Bars below shows the amount of time it takes by controller to have its performance affected and amount of time it takes to take down the controller using the DDOS attack. Bars below shows that controller without having any security is taken down after around 30 seconds and controller with security do not provide any SYN-ACK, so automatically prevents the attack and helps in preventing the DDOS attack.

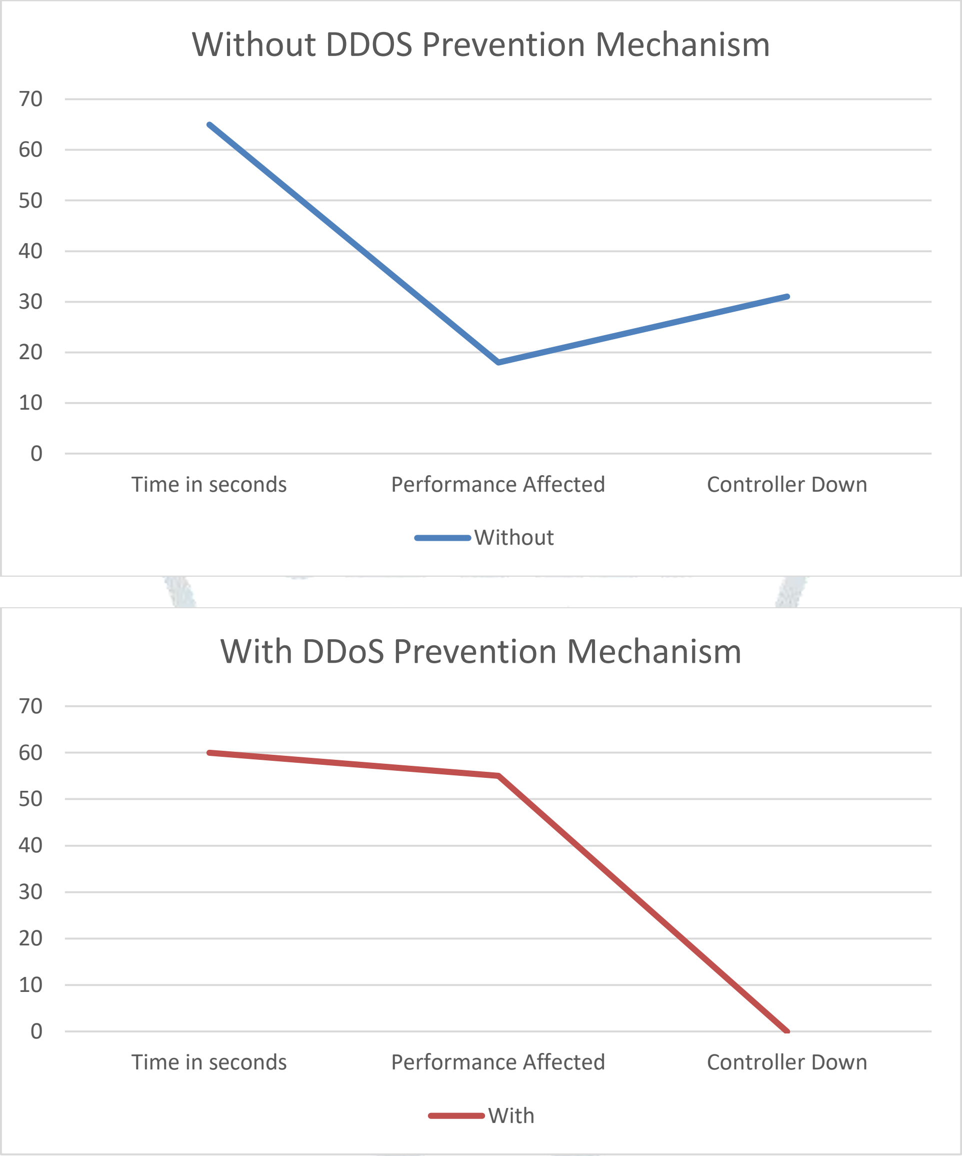


Figure7.6 :Results in bar graphs showing the time taken by controller in having performance affected and controller down

VIII.CONCLUSION

SDN is a network technology that has changed the way we design, manage and implement networks. It was the outcome of the clean slate project in Stanford University and soon it started to make strides to change the networks around the world and now it is one of the technologies which are acting as 5G enabler. It brings lots of benefits to the network industry both financially and performance wise. One of the biggest challenge that it has to deal with is the security, as there is a single point of failure and if hackers or intruders be able to compromise the controller, they ultimately hacks the whole network. Attackers can create a botnet and then perform a DDOS attack which can disrupt the whole Data Center network integrated with the controller. We can use linux inbuilt firewall i.e. IPTables to reduce the impact of DDOS attack by limiting the amount of traffic which has to be replied with any reply packet as in case of TCP SYN. This script can help controller in having a security in case of DDOS attacks without having any explicit security.

IX.FUTURE SCOPE

SDN is relatively new technology in the industry and a very hot research topic in network industry. Security is the major challenge that it faces at the moment with different concerns like DDOS which further divides into different methods of attack like TCP SYN Flood, UDP Flood, HTTP Flood, ICMP Flood etc. We have worked on TCP SYN Flood by using Linux Based firewall to prevent this issue. HTTP Flood is a very severe DDOS attack and in future, work can be carried on HTTP Flood Detection and Prevention using real controller like Opendaylight and devices with the help of Machine Learning.

REFERENCES

[1] L. Ertaul, K. Venkatachalam,"Security of Software Defined Networks (SDN)", The 2017 World Congress in Computer Science, Computer Engineering, and Applied Computing (CSCE'17), The 16th Int'l Conf on Wireless Networks (ICWN'17), July, Las Vegas, 2017.

[2] Zhaogang Shu & Jiafu Wan & Di Li & Jiaxiang Lin & Athanasios V. Vasilakos & Muhammad Imran(2016), "Security in Software-Defined Networking:Threats and Countermeasures", Springer Science+Business Media New York.

[3] Diego Kreutz, Fernando M. V. Ramos and Paulo Verissimo(2013), "Towards Secure and Dependable Software-Defined Networks", HotSDN'13, ACM.

[4] D. Kreutz, F. Ramos, P. Verissimo, C. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey,"Proceedings of the IEEE, vol. 103, no. 1, pp. 14-76, January 2015.

[5] Alexander Gelberger, Niv Yemini, Ran Giladi," Performance Analysis of Software-Defined Networking (SDN)"IEEE,2013.

[6] XU Xiaoqiong, YU Hongfang, and YANG Kun(2017), "DDoS Attack in Software Defined Networks: A Survey", ZTE COMMUNICATIONS.

[7] Nick Feamster, Jennifer Rexford, Ellen Zegura(2015), "The Road to SDN: An Intellectual History of Programmable Networks", Princeton, USA.

[8] OpenDaylight project, "https://www.opendaylight.org",2016.

[9] GitHub of OpenDaylight IntegrationProject,https://github.com/opendaylight/integration.

[10] Nicira. It's time to virtualize the network, 2012. http://nicira.com/en/network-virtualization-platform.

[11] NSF Guidelines for Planning and Managing the Major Research Equipment and Facilities Construction (MREFC) Account. http://www.nsf.gov/bfa/docs/mrefcguidelines1206.pdf, Nov. 2005.

[12]Open Networking Foundation. https://www.opennetworking.org/.

[13]Open vSwitch. openvswitch.org.

[14]Quagga routing software suite.

[15] Scott Shenker, Martin Casado, Teemu Koponen, Nick McKeown, et al. The future of networking, and the past of protocols. Open Networking Summit, 20:1-30, 2011.

[16] Open Networking Foundation. Software-defined networking: The new norm for networks. ONF White Paper, 2:2-6, 2012.

[17] Software defined networking, big switch networks.

[18] Software defined networking, microsoft.

[19] Radia Perlman, Anoop Ghanwani, Donald Eastlake 3rd, Dinesh Dutt, and Silvano Gai. Routing bridges (rbridges): Base protocol speci_cation. 2011.