ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 8, August 2019

Simulation of Flooding based DDoS Attack in Software Defined Networks

Rajni Samta[1], Satish Kumar[2]

Department of Computer Science, Himachal Pradesh University, Shimla, India

Department of Computer Science, Himachal Pradesh University, Shimla, India

ABSTRACT: This paper carries out the description of the methodology that has been carried out in order to simulate flooding based DDoS attack in the software Defined Network environment. Our aim is to understand the way in which DDoS attack can manage to tamper the Software Defined network and what vulnerabilities this network holds against such attacks. This study would be a building block for analysing various parameters for Software Defined Networks like throughput, round-trip-time and efficiency in terms of packet delivery.

KEYWORDS: Software Defined Network, DDoS Attack, Mininet, ICMP Ping attack, Hping3, Fragmentation.

I. INTRODUCTION

DDOS ATTACKS:
DDos attacks are very disastorous form of attacks as these have wide variety of ways to implement and it possess different categories like Flood attacks, Amplification attack, Protocol exploit attack, Malformed packet attack etc. Categorization of DDoS attacks are as follows [1]:

1. BANDWIDTH DEPLETION:
   i. Flood attack
   a) UDP
   b) ICMP
   ii. Amplification attack
   a) Smurf
   b) Fraggle
2. RESOURCE DEPLETION:
   i. Protocol exploit attck
   a) TCP SYN
   b) PUSH ACK
   ii. Malformed packet attack
   a) IP address
   b) IP packet options

MININET-SOFTWARE DEFINED NETWORK SIMULATION:
Mininet is an emulator to create a realistic virtual network consisting of virtual hosts, switches, links and controllers. It uses a command line interface. Various controllers like POX, NOX, floodlight etc. use different scripts such as python and java. Mininet is unable to run on a non-Linux environment. The topology that has been created is a tree topology with six components (four hosts, two switches) and a remote controller [2]. Here in this scenario the attacker is host one and is sending multiple ping requests towards the victim host three. The remote controller that we have used is a POX controller in a learning mode and the switches are open flow OVS switches [3], as shown in Figure1.

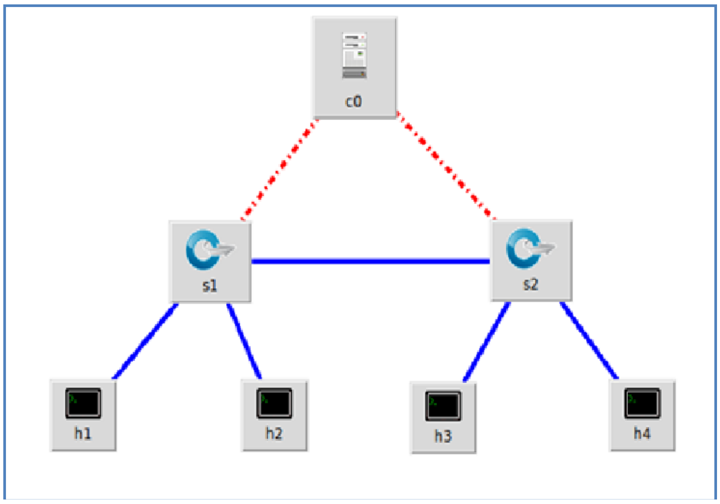


Figure1: Software Defined Network (Mininet).

ICMP (PING ATTACK):
It is an old form of attack called ping of death which causes a remote system to crash by sending just a malformed Internet Protocol IP packet. That entire attacker has to do is to create an ICMP (Internet Control Message Protocol) echo-request which is popularly known as Ping now a days. Normally Ping packets are used for figuring out round-trip time latency and number of hops of each packet. As Ping Packets are very light weighted these can also be used to check heart beat or status of any remote system or even used for checking self connectivity as well [4].

HPING3:
The tool that is used to generate the DDoS flooding attack is Hping3, which is an open source tool. It is capable of generating the required packets in a certain amount of rate and number. It is a command based tool and does not provide any graphical user interface [5]. Launching an attack from host one's xterm to clog the path of victim host three is shown in Figure2.

DOI: 10.15680/IJIRCCE.2019. 0708004

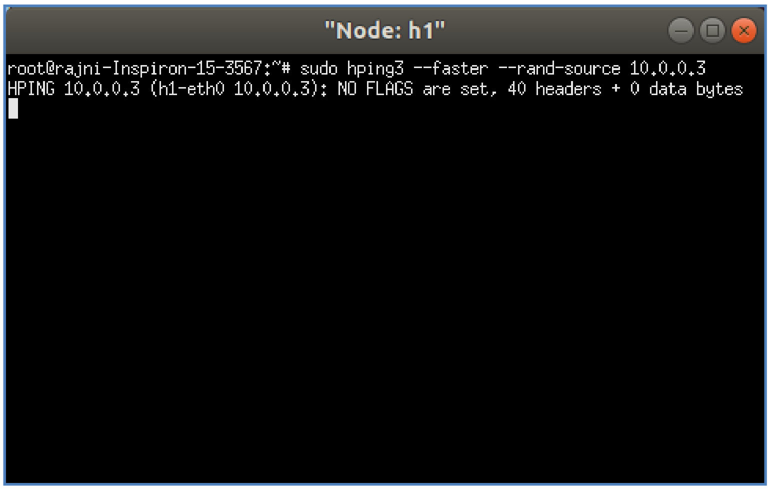
ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 8, August 2019



```
root@rajni-Inspiron-15-3567:~# sudo hping3 --faster --rand-source 10.0.0.3
HPING 10.0.0.3 (h1-eth0 10.0.0.3): NO FLAGS are set, 40 headers + 0 data bytes
```

Figure2: Launching DDoS Attack using Hping3.

II. RELATED WORK

Kumar, S. [6]. This paper has discussed that Ping attacks have been very dangerous in terms of shutting down the entire network as it attacks the DNS root Servers and has managed to bring down very popular websites such as EBay, ETRADE and Yahoo. This evaluation has been achieved by simulating the Ping attack traffic under controlled environment. This work has helped in understanding the impact that Ping attacks possibly could make on network.

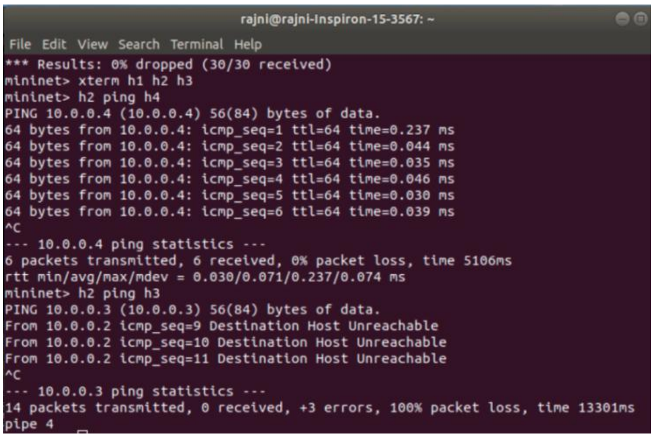
Moustis, D. et al. [7]. This paper contains the discussion related to bots in the field of distributed denial of service attacks and has discussed how large number of infected machines known as bots controlled by a bot master tends to take down a target service or tries to make any service in the network unavailable for genuine user. This paper not only discussed the impact of bots but also proposed some security control measures in order to check the effectiveness of these kinds of assaults.

Dong, P. et al. [8]. This paper specifically explains distributive denial of service attacks for the Software Defined Networks environment. As known already controller is the key feature for its uniqueness and weakness. Distributive denial of service attack while attacking the controller it may cause serious issues for the entire network as well. This paper has proposed detection methods are key features that reduce vulnerability against such security threats.

Zhang, P. Et al. [9]. This paper has described evolution in traditional networks having the same architecture, nothing has been changed so far until SDN, but just adding few more protocols or new topologies to make network faster and more reliable. Undoubtedly, traditional networks were at their best, but with the arrival of the era of clouds, where the organizational needs are differing day by day, service providers need to satisfy various network service requirements (like bandwidth, service quality, safety or reliability) for different users, which needs the network to be highly flexible, and virtual. Whereas, traditional way of networking failed to accommodate new features factors being: software and hardware are tightly coupled, and network protocols that are too complicated are integrated into the devices which are manufacturer-proprietary.

III. SIMULATION OF ATTACK

CONVERTING AN ICMP ECHO-REQUEST INTO DDOS ATTACK:
As shown in Figure 2, hping3 tool has generated massive ping packets towards host3 of the network, the victim host has gone unreachable (shown in Figure 3) from the moment it started receiving the ping packets from attacker.


rajni@rajni-Inspiron-15-3567: ~
File Edit View Search Terminal Help

```
*** Results: 0% dropped (30/30 received)
mininet> xterm h1 h2 h3
mininet> h2 ping h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=0.237 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=0.044 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.035 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=0.046 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=0.030 ms
64 bytes from 10.0.0.4: icmp_seq=6 ttl=64 time=0.039 ms
^C
--- 10.0.0.4 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5106ms
rtt min/avg/max/mdev = 0.030/0.071/0.237/0.074 ms
mininet> h2 ping h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
From 10.0.0.2 icmp_seq=9 Destination Host Unreachable
From 10.0.0.2 icmp_seq=10 Destination Host Unreachable
From 10.0.0.2 icmp_seq=11 Destination Host Unreachable
^C
--- 10.0.0.3 ping statistics ---
14 packets transmitted, 0 received, +3 errors, 100% packet loss, time 13301ms
pipe 4
```

Figure3: Victim Host unavailable.

WORKING OF THE PING ATTACK:
As we have discussed in the introduction section, that the IP Ping packets are light in weight, generally of size 65,535 Bytes, but attackers usually tend to send much larger packets than the limit. Well sending a larger packet is not an issue these days because of the concept of fragmentation. Usually the packets with much large size are fragmented in order to take care of MTU (Maximum Transmission Unit). Similar technique is adopted by an attacker to create large Ping packets and in massive amount. The goal is to make the victim host unreachable and it tends to eat up the memory buffers or internal memories at the receivers end. An illustration of what happens at the backend is shown in Figure4.

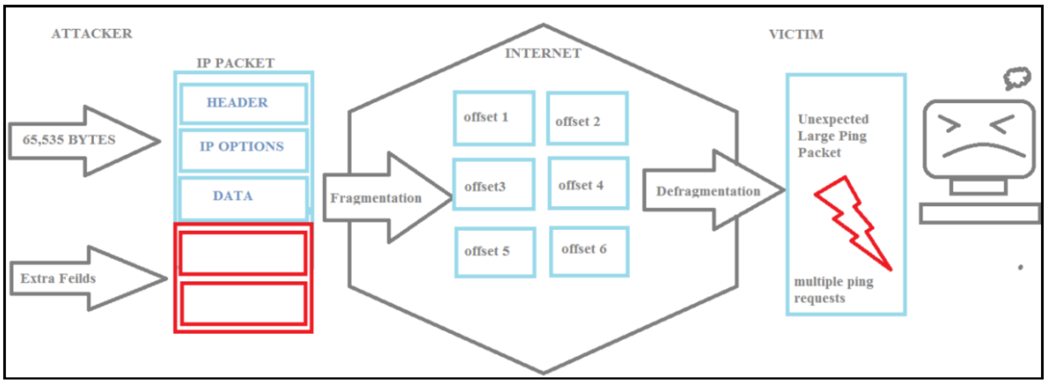


Figure4: Illustration of Ping Attack.

IV. CONCLUSION AND FUTURE WORK

We have described in detail various types of DDoS attacks and mainly focusing on the flooding based DDoS attacks (ICMP echo-request and reply). Describing the process of how this type of attack tries to fail the service of the network.

While dealing with flooding based attacks one problem that we faced is that these attacks try to clog the link of the network or a particular host which sometimes lead to crashing of the whole system and mitigation process fails to even operate on first place. Detecting a DDoS flooding attack is itself a challenge because sometimes it might go unspotted as well, like we never come to know why some particular component is not responding or participating in given communication. So in Software Defined Networks there are some controller based detection and mitigation techniques which make the process easier. This study helps in understanding the behaviour of Software Defined Networks while under DDoS attack and can be measured for various parameters of network performance.

REFERENCES

1. **J. Mirkovic**, and **P. L. Reiher,** “A Taxonomy of DDoS Attack and DDoS Defense Mechanisms”, Association of Computing Machinery, pp. 39-53, vol. 34, no. 2, April, 2004
2. Mininet Topologies at http://www.routereflector.com/2019/5/mini net-as-an-sdn-test-platform
3. Fernandez, Marcial. "Evaluating OpenFlow Controller Paradigms." In ICN 2013, The Twelfth International Conference on Networks, pp. 151–157. 2013.
4. Neumann Peter G. Denial-of-service attacks. ACM Communi-cations April 2000;43(4).
5. HPING3. http://www.hping.org/manpage.htm.
6. Kumar, S. (2006). PING attack – How bad is it? Computers & Security, 25(5), 332–337.
7. Moustis, D., & Kotzanikolaou, P. (2013). Evaluating security controls against HTTP-based DDoS attacks. IISA 2013.
8. Dong, P., Du, X., Zhang, H., & Xu, T. (2016). A detection method for a novel DDoS attack against SDN controllers by vast new low-traffic flows. 2016 IEEE International Conference on Communications (ICC).
9. Zhang, P., Wang, H., Hu, C., & Lin, C. (2016). On Denial of Service Attacks in Software Defined Networks. IEEE Network, 30(6), 28–33.
10. S.T. Zargar, J. Joshi, D. Tipper, and S. Member, “A Survey of Defense Mechanisms Against Distributed Denial of Device (DDoS)”, IEEE Communication Survey Tutorials, vol. 15, no. 4, pp. 2046–2069, 2013.
11. R. Wang, Z. Jia, and L. Ju, “An Entropy-Based Distributed DDoS Detection Mechanism in Software Defined Networking”, In Proceedings of IEEE Trustcom/BigDataSE/ISPA, pp. 310–317, 2015.