IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
PP 27-32

Security in SDN for IoT Enabled Attacks

Trupti Lotlikar
Phd student, Terna college eof Engineering, Terna Assistant Professor, Information Technology, Fr.CRIT, Vashi

Abstract: *In the past few years there has been a tremendous increase in network attacks especially the Denial of Service and Distributed Denial of Service attacks. The attacker tries to send a number of packets to the victim site so as to make it unavailable to provide any service. Nowadays DoS attacks can be performed using IoT enabled devices too. Traditional network architecture are unable to handle such attacks which gave rise to Software Defined Network(SDN) as a solution.Software defined networking is a network technology that provides central control over the network and thus manages network behaviour dynamically through software via open interfaces.SDN plays a role to retain the heterogeneity in networks and objects by integrating into them a control solution that interacts with the SDN controllers. Although SDN provides promising solutions to many security problem, it is also exposed to new attacks targeting the data plane, control plane or the link connecting the two planes. This paper aims at the integration of IoT with SDN technology, study of OpenFlow Protocol, different DoS attacks occurring on the layered SDN structure and the possible mitigation strategies to weaken the impact of those attacks on the network. For this purpose, I will be using mininet simulator, OpenVSwitch(OVS) and Floodlight SDN controller.*

Keywords: *IoT, SDN, Open Flow, attacks, mininet.*

I. Introduction

With the rapid growth of the Internet over the last two decades, the number of attacks on the Internet services has increased rapidly. One such example involves disrupting the service provided by a network or server either by crashing the systems, sending some packets that exploit a software vulnerability or by sending a large amount of useless traffic to collapse the resources of the service. This kind of attack is nothing but Denial of Service (DoS) attack, or a Distributed Denial of Service (DDoS) attack if it is launched by multiple hosts. Previously, hackers relied on large networks of computers to execute DDoS attacks. But with the rise of Internet of Things(IoT), performing such attacks have become more simpler and highly destructive as hacker just needs nominal control over a lot of internet-connected computers, enough to overwhelm their target when it tells the network to visit a certain website.

To handle such attacks is visibly impossible using traditional networks because switches in traditional networks does not have programmability as rules cannot be changed dynamically. Whereas Software Defined Network(SDN) is comparatively suitable, as it is a framework that allows network administrators to automatically and dynamically manage and control large number of network devices, services, topology, traffic paths and packet handling (quality of service) policies using high- level languages and APIs. Using OpenFlow Protocol, communication between the controller and network devices can be managed in a better way. It is a set of specifications used for manipulating the switch's configuration state as well as receiving certain switch events. This paper discusses how attacks are performed using IoT devices. And how it can be mitigated in SDN using OpenFlow Protocol. The rest of the paper is organized as follows. Section II describes various technologies used. Section III defines the Research Idea. Section IV elaborates the proposed system. Section V explains the mitigation strategies needed to protect the system from attacks. Finally Section VI gives the conclusion.

II. SDN Overview

With the increasing advent of IoT(Internet of Things) in the recent times, connecting and operating devices have become much simpler. By assigning IP address to the objects, data can be collected and transferred through various devices over a network. Decisions are taken based on the interaction of the objects(with embedded technology) between internal and external states.

Inspite of all this, new complexities have been imposed on both networking and internetworking areas. IoT indicates the interconnection of several heterogeneous networks, the objects that compose them, the environments they are running, the protocols they use and the different objectives they have. This challenge could be solved using SDN protocol.

At the same time, various attacks using IoT devices are also posing huge threats. If proper mitigation methods are not implemented, then it may sometimes lead to huge data loss or poor networking services. As

already known, IoT is all about connecting and networking devices, which means that all of those devices, whether it is a brand new connected refrigerator or vehicle, are creating a new entry point to the network and therefore posing an increasing security and privacy risk. Poor security on many IoT devices makes them soft targets and often victims may not even know they have been infected. Most commonly observed attack is DOS attack, which happens when a service that would usually work is unavailable.

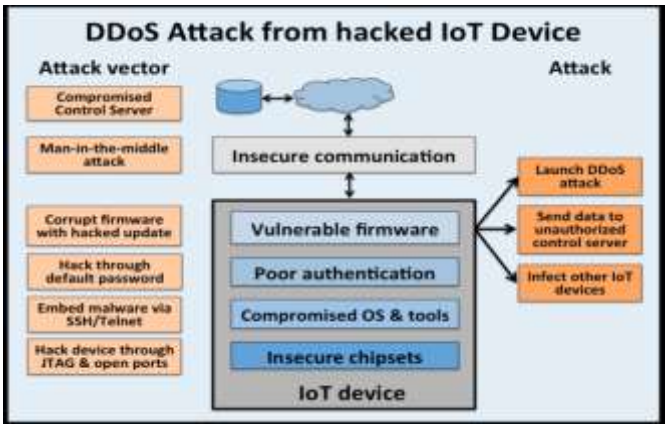

Fig. 1: Denial of Service attack using IoT device.

Fig 1 depicts DoS attack using IoT device. There can be many reasons for unavailability, but it usually refers to infrastructure that cannot cope due to capacity overload. In DDoS attack, a large number of systems maliciously attack one target. This is often done through a botnet, where many devices are programmed (often unbeknownst to the owner) to request a service at the same time.

Another important concept used is Software Defined Network (SDN), which attempts to build a computer network by separating it into two systems- Control Plane and Data Plane. However the functioning of SDN is depicted in fig 2. There are three layers. The Data Plane also known as Infrastructure layer, constitutes network devices like switches, routers ,etc. It is responsible for forwarding or dropping the incoming packets according to the flow table configured from control plane through southbound protocol.

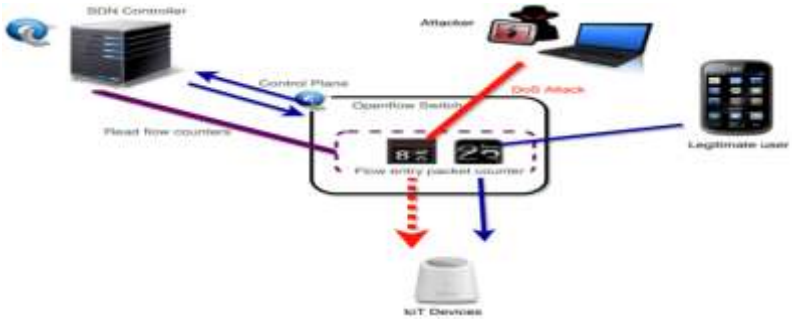

Fig. 2: SDN architecture

Control Plane consists of Controller which provides performance and fault management. The "brain" of the Control layer is an SDN Controller. It typically handles configuration management of the SDN compliant devices and understands the network topology. Loaded with these details, the controller can process connection requests based on desired requirements such as QoS levels, etc. Controller can also perform link management between the devices and interconnection of applications. The Control Plane configures the connection paths or flows into the data plane through the use of control protocol. The control protocol is used by a controller in software defined network to perform important functions such as connection setup. The uppermost layer could be composed of applications managing and securing the underlying network. The applications could be running on the Controller or can communicate through the northbound API of the controller. Due to the centralized knowledge of the network, they are able to gain information about the whole network from the controller. On the basis of this knowledge, applications can configure and upload records into flow tables of switches in data plane through the northbound API of the controller.

The protocol used for managing SDN controller is known as OpenFlow Protocol viz. an Open API that provides a standard interface for programming the data plane switches. It is used for remotely controlling the

forwarding table of a switch or router and is an element of SDN. OpenFlow based controllers will discover and maintain an inventory of all the links in the network and then will create and store all possible paths in entire network. OpenFlow protocol can instruct switches and routers to direct the traffic by providing software based access to flow tables that can be used to quickly change the network layout. The OpenFlow architecture consists of three basic concepts. (1) The network is built up by OpenFlow-compliant switches that compose the data plane; (2) the control plane consists of one or more OpenFlow controllers; (3) a secure control channel connects the switches with the control plane. Fig. 3 depicts the basic packet forwarding mechanism with OpenFlow in a switch.

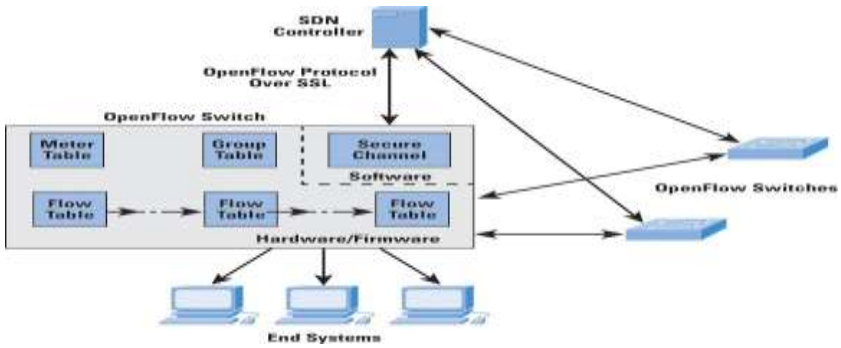

Fig. 3: Switching Mechanism

Mechanism begins when a packet is first received by the switch. The packet's header is parsed & matched against the flow table. If a flow table entry is found, only then the packet is allowed to enter. But if there's ample of entries, packets are matched based on prioritization, i.e. the most specific entry/ highest priority is selected. Then the counters of the flow table are updated by the switch, which then performs the mentioned action, eg. forwarding of packet to port, etc. If no references of packet are found, the switch notifies the controller and buffering occurs whenever switch is ready for buffering. Using PACKET-IN message, encapsulation of unbuffered packet or first bytes of buffered packet takes place. Once the controller receives the message, it identifies correct action for the packet and installs entries in the requesting switch. Thus controller does the main function of setting up the path for the packet in the network by modifying flow tables of switches.

III. Research Idea

This research is undertaken to solve the problem of IoT-enabled Dos and DDoS attacks on SDN. The idea is to use techniques which work as a solution to overcome these attacks in Software-Defined Networks. We use Mininet tool to create basic SDN which will also have an SDN controller.

We would perform Dos and DDoS attacks on this network through IoT-enabled devices. They will have an IP address which will be used to send blank packets to the target server to bring it down. We will then apply certain measures to counter act on this.

IoT-enabled devices include: Wireless Camera, Electric Coffee Machine, Refrigerators, Air-Conditioners. This will be done mainly to observe how the traditional network handles it and how the Software Defined Network will handle it. Based on this study we will get to know what all steps need to be considered to make a network durable.

The performance measure would be how effective the IoT-enabled devices are to bring down a server in the network and to check the level of mitigation.

In the paper, I provide the technique to prevent SDN (software defined network) network from the DDoS attacks via the IoT-enabled devices and also addressing the various detection and mitigation strategies of those DDoS attacks The performance measure in would be how effective the IoT-enabled devices are to bring down a server in the network and to check the level of mitigation.This paper aims at providing a solution to prevent DoS or DDoS attacks on SDN. The attacker could be either an IOT-Enabled device or an attacker within the network. SDN is implemented by segregating the control and data plane, the former holding the logical part and the latter performing the tasks accordingly. Both the Control plane and Data plane in SDN communicate using a protocol called as OpenFlow specified by the Open Networking Foundation (ONF). Thus, SDN is also known as smart network.

The probable attacks on the control and data plane can cause buffer overflow, control channel congestion, controller resource saturation and flow table overflow. By using several software and creating proper rules within controllers, I intend to achieve all the mentioned objectives and provide appropriate solutions. A Virtual network with all its basic components can be created using a software known as Mininet. Floodlight or PoX [5] is a SDN controller that can be used and OpenVSwitch (OVS) is used to imitate

OpenFlow protocol. Thus, our solution helps in detection of DoS and DDoS threats on both data plane and control plane and implements necessary actions.

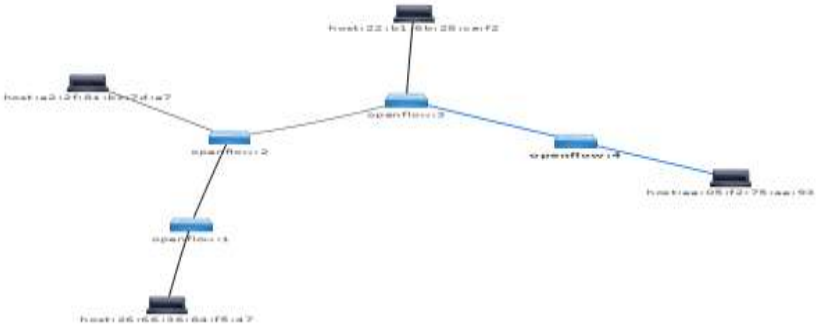

Fig 4. Topology depicted with Mininet

IV. Proposed system

The research proposal can be summarized as follows:-

1. The first step towards this is to create a software defined network virtually. This is done by using the software known as Mininet. Based on our requirements, we can create the components of the network like switches, hosts and controller. By default, Mininet uses PoX controller which is based on python.
2. The next step is to connect a wireless or wired device to the internet with the help of Internet of Things. The device used in this project will be a wireless camera. Any device connected to the internet is bound to get a logical IP which can be used to perform Dos and DDoS attacks.
3. The next phase is an observation phase where we observe how the IoT device connected to the internet is able to attack and up to what extent is it successful in doing so. We can also get to see how software defined networks handle DoS attacks as compared to the traditional networks.
4. The next step is to implement measures into the software defined networks such that it prevents these types of attacks to occur to a certain extent.

Different DoS attacks which can be performed are:

1. Ping of Death- In order to find if the network resource is available or not, ping command can be used. It works by sending small data packets to the network resource. The ping of death takes advantage of this and sends data packets above the maximum limit (65,536 bytes) that TCP/IP allows. TCP/IP fragmentation breaks the packets into small chunks that are sent to the server. Since the sent data packages are larger than what the server can handle, the server can freeze, reboot, or crash.

2. Buffer overflow- A buffer is a temporal storage location in RAM that is used to hold data so that the CPU can manipulate it before writing it back to the disc [1]. Buffers have a size limit. This type of attack loads the buffer with more data that it can hold. This causes the buffer to overflow and corrupt the data it holds. An example of a buffer overflow is sending emails with file names that have 256 characters.

3. SYN attack- SYN is a short form for Synchronize. This type of attack takes advantage of the three-way handshake to establish communication using TCP. SYN attack works by flooding the victim with incomplete SYN messages. This causes the victim machine to allocate memory resources that are never used and deny access to legitimate users[1].

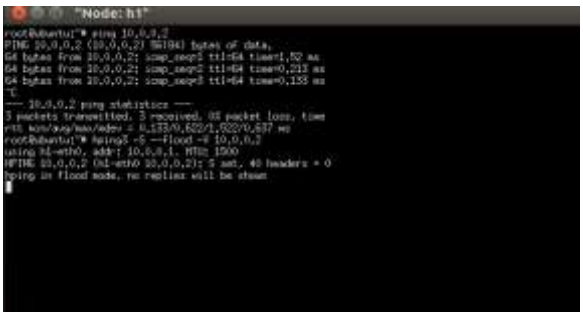

Fig 6: Snapshot of attacker node performing DoS attack

As shown in Fig 6, virtual network is created using Mininet and accordingly several components like switches, hosts and controller are created. An IoT device once connected to the internet, depicted by host h2 with ip address 10.0.0.2 is then launched by hping3 DoS attacks by flooding network with packets by attacker h1.

Once the attack progresses, it is very important that the defence mechanism differentiates between the real attack and normal legitimate traffic. One way is by detecting the pattern where incoming traffic is checked if it matches known signatures of the database. This is efficient unless there's a new attack pattern which is not present in database. Another way is Anomaly based detection which identifies malicious activity in a network by detecting anomalous network traffic patterns.

V. Mitigation strategies

This paper provides some mitigation techniques against the Denial of Service (DoS) attacks that has been mentioned in this paper.

One of the mitigating strategy against DoS attacks is by using the Rate Limiter. A switch is introduced along with the controller, when the flow remains inactive and a hard timeout, which is triggered at the timer expiration. When either of these timers expire, the switch removes the corresponding flow entry from its flow table and sends an OFPT FLOW REMOVED message to the controller [7]. If the switch receives a larger number of packets together in a particular duration, then the buffer gets filled and remaining incoming packets are not processed therefore the switch becomes unable to forward traffic from new flows.

Here we are using packet flow controlling methods to mitigate an attack on the controller bandwidth by enforcing a rate limit on the number of packets sent to the controller. A Switch provides Meters, the meter basically acts as a Rate limiter [7]. A meter table consists of meter entries, defining per-flow meters. Per-flow meters enable OpenFlow to implement various simple QoS(Quality Of Service) operations, such as rate-limiting.

This mitigation technique has a Rate Limiter which is basically implemented to prevent incoming large set of packets by defining a threshold,it limits the packet receival rate by controlling and dropping the traffic from the external attacker if an attack happens. After the control is effective it is then released so that packets are released from the blocking state and when a new attacker attacks a new event will be generated triggering a new control.

Another method of mitigation is by maintaining flow table with the aim to collect information from the received and stored packets through the DoS attacks 31ccurred, whose target is to bring down the network by flooding it with excessive traffic. The bandwidth of the channel between the switch and the controller gets overloaded which then results in the congestion of packets in that channel.

Another solution is to assign hard timeouts for each flow rule that are to be provided to the flow of packets. If the incoming flow of packet is malicious, this mechanism assigns its forwarding rules a high timeout. This is to ensure that the same flow rule does not trigger many communications between the switch and the controller. Yet another solution for this issue is to aggregate flow rules entries of malicious flows at a particular switch. This is done so that the rules of flows corresponding to a same source do not require to be written again for every other packet in the TCAM table allowing more rules to adjust. The control plane can have the logic related to two types of flows which are the Legitimate flow and Malicious flow [6].

VI. Conclusion

This paper titled "IoT-enabled DoS attack mitigation strategy in software defined networks" makes use of two main technology viz. Internet of Things (IoT) and Software Defined Networks (SDN). It helps us to study the various ways in which DoS attack can be performed using IoT-enabled device. It also gives us a brief comparison on how traditional networks handle these attacks and how well the Software Defined Networks handle them. Moreover, based on the comparison, the paper also assists in implementing necessary actions so as to mitigate the attacks to a certain extent.

References

[1]. A. García de la Villa, "Distributed Denial of Service Attacks defenses and OpenFlow: Implementing denial-of-service defense mechanisms with software defined networking", *Core.ac.uk*, 2017. [Online]. Available: https://core.ac.uk/display/80713323. [Accessed: 25- Oct- 2017].

[2]. Ferguson, P. Network ingress filtering: Defeating denial of service attacks which employ ip source address spoofing. Amaranth Network Inc. (2000).

[3]. Peng, T., Leckie, C., and Ramamohanarao, K. Protection from distributed denial of service attacks using history-based ip filtering. In Communications, 2003. ICC'03. IEEE International Conference on (2003), vol. 1, IEEE, pp. 482–486.

[4]. Chang, R. K. Defending against flooding-based distributed denial-of-service attacks: A tutorial. Communications Magazine, IEEE 40, 10 (2002), 42–51. Bellovin, S. M., Leech, M., and Taylor, T. Icmp traceback messages. Internet Engineering Task Force, Marina del Rey, Calif (2003).

[5]. NOX. POX Controller. http://www.noxrepo.org/pox/about-pox/. [Online; accessed 24-Oct-2017].

[6]. L. Dridi and M. Zhani, "SDN-Guard: DoS Attacks Mitigation in SDN Networks - IEEE Conference Publication", *Ieeexplore.ieee.org*, 2017. [Online].Available: http://ieeexplore.ieee.org/document/7776605/. [Accessed: 24- Oct- 2017].

[7]. M. Kuerban, Y. Tian, B. Huebert, D. Poss, Q. Yang and Y. Jia, "FlowSec: DOS Attack Mitigation Strategy on SDN Controller - IEEE Conference Publication", *Ieeexplore.ieee.org*, 2017. [Online]. Available:http://ieeexplore.ieee.org/document/7549402/.

[8]. Braga R, Mota E, Passito A (2010) " Lightweight DDoS flooding attack detection using NOX/OpenFlow.: IEEE 35th Conference on Local Computer Networks (LCN)
[9]. R. Kloti, V.Kotronis, P.Smith, "OpenFlow: A security Analysis,"2013.
[10]. Po-Wen Chi, Chein-Ting Kuo, He-Ming Ruan, Shih-jen Chen and ChiN-Laung Lei, "An AMI Threat Detection Mechanism Based on SDN Networks" in Eight international conference on emerging security information, system and technologies, Taiwan .
[11]. Yogita Hande and Aishwarya Jadhav "Software defined networking with Intrusion Detection System" in International Journal of Engineering and technical Research (IJETR) Volume 2, issue-10, October 2013.
[12]. Sandra Scott-Hayward, Member IEEE, Sriram Natarajan and Sakir Sezer, Member IEEE "A Survey of Security in Software Defined Networks" IEEE communication survey and tutorials, Vol 18,No.1. First quarter 2016.
[13]. Mohan Dhawan, Rishabh Poddar ,Kshiteej Mahajan, and Vijay Mann."SPHINX: Detecting Security Attacks in Software-Defined Networks." In Proceedings of the 22th Annual Network and Distributed System Security Symposium (NDSS'15), February 2015.