Distributed Denial of Service Attack Detection Using SDN

Project Group-2

Project Description & Objective:

Software-Defined Network, in short, an SDN is a three-layer architecture, which consists of the application layer, the control layer, and the infrastructure layer. The below-given figure-1 illustrates the architecture of the SDN. This type of framework provides a centralized network management system to the network admins to easily configure the network. It also provides the administrators with a programmable UI to develop network management tools, network security, and network logging. The controller layer in the SDN architecture is responsible for controlling the entire network, giving visibility of the network, and tracking the network resources to those who manage it. The infrastructure layer consists of the network end devices involved with forwarding packets to other network devices if the information is present in the data plane. If not present, it queries the SDN to update its data plane, and then it forwards the packet. The top layer is the application layer; this is where the applications are developed to monitor, control, and create apps related to the network's other functionalities. All these advantages of SDN also increase the attack surface and vulnerabilities related to it. The impacts related to it are more severe than the conventional networks we are using right now. Our objective in this project is to gain in-depth knowledge regarding the workings of Software-Defined Networking (SDN).

We also learned about attacks happening at different layers of SDN. In this project, we mainly focused on detecting Distributed Denial of Service (DDoS) attacks happening in our SDN infrastructure. DDoS is a Denial of Service (DoS) attack using multiple botnets attack systems. We will create our network topology using Mininet, a tree topology with n switches, and network packets will be created using Scapy, a python-based tool. We will run our controller with Pox, a Python SDN controller. We will typically detect early, and the central theme behind this is entropy, which is the measure of disorderliness in a system. When a DDoS attack occurs in the system, there will be a sudden increase of randomness that will decrease the system's entropy. This project will use the SDN controller for DDoS attack detection, and an early detection mechanism should be a lightweight and high response. This increased response time will help the controller regain control during the attack.

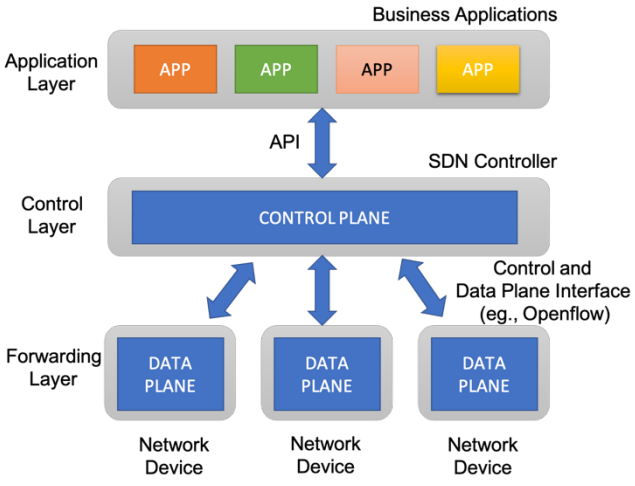

Figure 1 SDN 3-Layer Architecture.

Project High Level Design:

In order to simulate our project, we will be using the following tools to generate network topologies and DDoS attacks.

1. **Mininet:** We will use it to create a tree topology consisting of n switches. It is a simple network emulation tool used for SDN. By Mininet deployment of large topologies can be done with limited resources on a single Virtual Machine. An extensible Python API is provided by Mininet, which enables for network creation and simulations.
2. **Pox:** This comes built-in with Mininet when it is installed. Pox provides a framework in SDN for connecting with switches using OpenFlow. This also provides a lightweight, fast, and designed in such a way that the customized controllers can be built on it. The developers who generally work with the Pox can use it to create an SDN controller using Python.
3. **Scapy:** This is a python-based library, which will be used to generate traffic to and from the SDN controller. This is an excellent tool for attacking, scanning, generating, and sniffing packets. UDP packets are generated by using Scapy and spoof the source IP of the packets.
4. **OpenFlow:** This is a communication protocol within the SDN architecture, which assists the SDN controller in determining the path of the network packets across switches. This parting of forwarding planes and control planes helps in managing complex traffic. This allows switches from multiple vendors to be remotely managed using a single protocol.
5. **SSH:** We will use SSH in order to connect Mininet machine from other local machines.
6. **Topology:** In this project we will use tree-based topology to execute our attack and mitigation.
7. **VirtualBox and VMWare:** VirtualBox and VMWare to install Mininet rather than installing full-fledged machine, and other components. We can use both on the basis of our team's expertise.
8. **Python:** The python script is to monitor entropy of the network and also keeps track of the switches and the number of packets received on each of the ports.

Test Cases:

We have listed the following test cases and possibly more will be added as we do further experimentation.

Test Cases	Description
1	First, we will create a topology and test the connectivity of the nodes using the pingall command.
2	Second, we will generate a normal traffic on our SDN network to find the threshold for usual traffic behavior.
3	Third, after generating random normal traffic we will be able to see a list of values for entropy and find the threshold.

4	Finally, Simulate the attack on the target host and we will see a potentially large decrease in the value of entropy, which means an increase in the randomness. This typically means that an attack is happening on the target host.

References:

[1] Mousavi, S. and St-Hilaire, M., 2017. Early Detection of DDoS Attacks Against Software Defined Network Controllers. *Journal of Network and Systems Management*, 26(3), pp.573-591.

[2] Hong, S., Xu, L., Wang, H. and Gu, G., 2021. *Poisoning Network Visibility in Software-Defined Networks: New Attacks and Countermeasures*.

[3] Velrajan, S. and Velrajan, S., 2021. *SDN Architecture*. [online] Tech.in | 5G, SDN/NFV & Edge Compute. Available at: <https://www.thetech.in/2012/12/sdn-architecture.html> [Accessed 26 March 2021].

[4] Open-Source Routing and Network Simulation. 2021. *Using the POX SDN controller*. [online] Available at: <https://www.brianlinkletter.com/2015/04/using-the-pox-sdn-controller/> [Accessed 26 March 2021].

[5] En.wikipedia.org. 2021. *OpenFlow - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/OpenFlow> [Accessed 29 March 2021].

[6] Open Networking Foundation. 2021. *MININET - Open Networking Foundation*. [online] Available at: <https://opennetworking.org/mininet/> [Accessed 29 March 2021].

[7] Infosec Resources. 2021. *What Is Scapy? - Infosec Resources*. [online] Available at: <https://resources.infosecinstitute.com/topic/what-is-scapy/> [Accessed 29 March 2021].